Proceedings

DoD Database Colloquium '95

"Emerging Technology for Database Interoperability and Data Administration"

19951026 060

San Diego Princess Hotel
San Diego, California
August 28 - 30, 1995

AFCEA

Proceedings

DRAFT SF 298

1. Report Date (dd-mm-yy)	2. Report Type	3. Dates covered (from... to)
08-95	PROCEEDINGS	

4. Title & subtitle	5a. Contract or Grant #
DOD DATABASE COLLOQUIUM '95	
	5b. Program Element #
6. Author(s)	5c. Project #
	5d. Task #
	5e. Work Unit #

7. Performing Organization Name & Address	8. Performing Organization Report #
THE ARMED FORCES COMMUNICATIONS AND ELECTRONIC ASSOCIATION 4400 FAIR LAKES COURT FAIRFAX VA 22033-3899	
9. Sponsoring/Monitoring Agency Name & Address	10. Monitor Acronym
DEFENSE INFORMATION SYSTEMS AGENCY-CENTER FOR SOFTWARE	11. Monitor Report #

12. Distribution/Availability Statement UNCLASSIFIED/UNLIMITED

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

13. Supplementary Notes

14. Abstract THIS IS THE PROCEEDINGS FROM THE DOD DATABASE COLLOQUIUM '95. THE THEME OF THE COLLOQUIUM WAS "EMERGING TECHNOLOGY FOR DATABASE INTEROPERABILITY AND DATA ADMINISTRATION". IT WAS HELD AUGUST 28-30, 1995, IN SAN DIEGO, CA.

DTIC
SELECTE
OCT 27 1995
S D
F

15. Subject Terms CIM (COLLECTION)

Security Classification of			19. Limitation of Abstract	20. # of Pages	21. Responsible Person (Name and Telephone #)
16. Report	17. Abstract	18. This Page			
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	Unlimited		LAWRENCE D PIERCE AFCEA 800-336-4583

CONFERENCE PROCEEDINGS

DoD Database Colloquium '95

August 28-30, 1995
Princess Hotel
San Diego, California

Accesion For
NTIS CRA&I ☒
DTIC TAB ☐
Unannounced ☐
Justification .....
By .....
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

Sponsored by:

The DoD Database Colloquium Coordinating Committee

In Cooperation With:

Defense Information Systems Agency-Center for Software (DISA/CSW)

National Security Agency-Central Security Service (NSA/CSS)

Naval Command and Control and Ocean Surveillance Center RDT&E Division

Naval Computer and Telecommunications Command

Naval Security Group Command

In Coordination with and Administered by:

The Armed Forces Communications and Electronics Association

The Armed Forces Communications and Electronics Association



4400 Fair Lakes Court
Fairfax, Virginia 22033-3899
USA

Telephone: (703) 631-6128 or (800) 336-4583
Facsimile: (703) 631-6133
E-Mail: bafceaprg@aol.com

This book cannot be copied in whole or in part without permission of the publisher

Printed in the United States of America

FOREWARD

Welcome to Database '95, DoD's twelfth and AFCEA's fifth Database Colloquium. The driving purpose of the Colloquium remains as always -- to enhance the professional knowledge of those participating, and to foster an ethical and mutually beneficial exchange of data management technology information between government, industry and academia. You are a welcomed and essential participant in this forum.

The papers included within these pages should provide you with sufficient detail on the subject matter to select those you want to know more about, or to make reference to when facing the challenges addressed by those papers. We regret that, in a few cases, authors may not have been able to receive clearance granting their papers inclusion in these proceedings; however, this is an unavoidable consequence of the different themes included in the conference, some of which may involve security, economic, strategic, or proprietary sensitivity. Additionally, there are usually some authors who simply do no forward their papers in time for inclusion. This too is probably unavoidable -- the result of the "busyness" of professional people, like you, who often have more to do than can be accomplished in a given time.

One copy of these Colloquium Proceedings will be provided to each registrant of DoD Database Colloquium '95. Requests to purchase additional copies should be referred to AFCEA, Attention: Programs Department, 4400 Fair Lakes Court, Fairfax, VA 22033-3899.

One final note: Views expressed by the authors are their own and do not necessarily reflect those of AFCEA, the United States Government or its agencies, or the participated corporations and institutions.

So, enjoy your time at Database '95. Talk to the authors and to each other. Be a player, and you will find it three days extremely well spent.

Lawrence D. Pierce, LTC (USA), Ret.
Colloquium Manager and Proceedings Editor

TABLE OF CONTENTS

PAGE

MONDAY, AUGUST 28, 1995

Keynote Address: Guest Presenter:
Diann L. McCoy Paper Not Requested.
Deputy Commander, Center for Software, DISA

Featured Address: 1
"The Application of Object Technology for Data Management"
Dr. Bhavani Thuraisingham, The MITRE Corporation

Panel: *"DoD Data Standardization Case Study"* Papers Not Requested.
Moderator: Rebecca Wade
Dept. of Navy Data Administrator
Naval Information Systems Management Center

Papers:

The 1995 Joint Warrior Interoperability Demonstration *
Ronald Elliott, Headquarters-U.S. Marine Corps

Defense Information Infrastructure: Data Migration Tasks, Techniques, and Solutions 3 included
Philip Cyanka, DISA-Center for Software

Use of Data Modeling to Coordinate Three Separate but Related Migration Systems ~~Available~~ but not numbered
Joyce Wineland, Office of Naval Intelligence

Defense Modeling and Simulation Data Administration 15
Dr. Chien Huo, Defense Modeling & Simulation Office
Ms. Iris Kameny, The RAND Corporation

COMNAVSECGRU Data Administration and Data Standardization 27
Duane L. Waggoner, Commander Naval Security Group

Achieving Joint Interoperability through Information Standards 37
LTC John D. Burke, USA, DISC4, HQ Dept. of Army

Data: The Critical Enabler for Managing Information Technology Assets in DoD 53
Andrew Verga, Wizdom Systems Inc.
Betsy Appleby, Defense Information Systems Agency

Incorporation of External Data Standards into the DoD Standardization Initiative 71
Ann W. Woody, DISA - Center for Software
Neal A. Levene, CSP, Vector Research, Inc.
Dave A. Paolicelli, Vector Research, Inc.
L. Tobias Klauder, Vector Research, Inc.

TUESDAY, AUGUST 29, 1995

Featured Address: Paper Not Requested.
The Management Dimension of Data Administration
Ms. Belkis Leong-Hong
Deputy Assistant Secretary of Defense (C^3I Plans and Resources)

Papers:
Object-Oriented Technology for Integrating Distributed Heterogeneous Database Systems 79
Dr. Marion G. Ceruti, NCCOSC RDT&E DIV 4221
Dr. Magdi N. Kamel, Naval Postgraduate School
Dr. Bhavani M. Thuraisingham, The MITRE Corporation

Database Integration Using NWTDB Procedures 99
R. Gressang, SWL Division, GRC International, Inc.
G. Michaels, SWL Division, GRC International, Inc.
E. Harris, SWL Division, GRC International, Inc.
J. Mathwick, SWL Division, GRC International, Inc.
J. Lu, SWL Division, GRC International, Inc.

Data Interoperability Between C^3I Systems 107
Scott A. Renner, The MITRE Corporation
Arnon S. Rosenthal, The MITRE Corporation
James G. Scarano, The MITRE Corporation

A Model for Information Retrieval from Heterogeneous Sources 117
Major Anthony Ruocco, USA
Dr. Ophir Frieder, George Mason University

Implementing Standing Requests for Information Over Distributed Heterogeneous Data Sources 129
Dr. Kenneth Smith, The MITRE Corporation
Patricia L. Carbone, The MITRE Corporation
Michael S. V. Turner, The MITRE Corporation

Envoy: Successful Multi-Agency Deployment of Heterogeneous Data Access 141
Dan Stickel, Delfin Systems
Tom Hillman, Delfin Systems
Larry Safran, Delfin Systems
Jerry Beersdorf, Delfin Systems

Data Sharing, Interoperability and Standardization
Peter Valentine, U.S. Army Electronic Proving Ground

A Technology Survey of Heterogeneous Data Access Across Multiple Data Types 155
Dr. David D. Mattox, The MITRE Corporation
Patricia Carbone, The MITRE Corporation
Dr. Marcia Kerchner, The MITRE Corporation
Ruth Hildenberger, The MITRE Corporation

Options for Object-Oriented Persistence 167
Dr. Marco Emrich, Cincom Systems, Inc.

U.S. Army Artificial Intelligent Center Integrated Database (AICIDB) System 173
MAJ Leonard Tharpe, U.S. Army Artificial Intelligent Center
CPT(P) Dionysis Anninos, U.S. Army Artificial Intelligent Center

Final Report of the DBSSG Predictable Real-time Information Systems Task Group 185
Dr. Paul J. Fortier, University of Massachusetts-Dartmouth
Donna Fisher, NCCOSC
David K. Hughes, Dbx inc.
Mayford Roark, Martin Marietta

Strategy As A Leading Edge To Creative Relational Database Management 197
Jan-Marie Esch, County Sanitation Districts of Orange County

Using Expert System Technology to Standardize Data Elements 205
Jennifer Little, Amerind, Inc.

included
~~Available~~ but not numbered

Standardized Metadata and Metadata Capture Tools for Migration to New Operational Systems and Development of DSS Infrastructures 219
Duane Hufford, American Management Systems

Managing Change Within Federated Database Schemas 241
Christopher J. Bosch, The MITRE Corporation

Natural Language Generator based on Database Modeling 251
1Lt Lee S. Waldron, USAF, Automate Communications Systems

Message-Oriented Middleware (MOM): A Key Technology for the Successful Deployment of Distributed Client/Server Information Systems 297
Gail V. Quigley, AT&T Global Information Solutions

Phillips Laboratory's Technology Transfer Database 311
Andrea E. Gleicher, USAF Phillips Laboratory
Lila A. Hicks, The Aerospace Corporation

Data Base Tools in Support of the Joint Modeling and Simulation Community *
Michael K. Hopkins, HQ United States Central Command

Database Replication and Synchronization in the Global Command and Control System 319
Ron Harris, SRA Technical Services Corporation

The U.S. Army Corps of Engineers Data Encyclopedia A Foundation for Interoperability 331
Stephen Vandivier, Avanco International, Inc.

Document Management and Production with Relational Database Management Systems 341
Carter M. Glass, Rapid Systems Solutions Inc.

CIM/EI Data Metrics 351
Pamela Piper, DISA-Center for Software

Utility of Data Mapping 359
Lynn Henderson, Defense Information Systems Agency
Feliza Kepler, Data Networks Corporation

Building Secure Applications 367
Jess Worthington, Informix Federal

Meeting the Warfighter's Command and Control Into Base Needs *
Lt. Col. K.D. Boyer, Jr., USAF
Joint Command and Control Warfare Center

Depot Maintenance Management Information System (DBBIS): Proof-of-Concept Data Quality Project *
Alta Paul, DISA-Center for Software

DBSism: A Tool for Predicting Database Performance 409
Mike Lefler, PRC Inc.
Mark Stokrp, PRC Inc.

WEDNESDAY, AUGUST 30, 1995

Papers:

Improving Security in Multi-Level Database Management Systems Through the Use of Query Modification 417
Dr. Michael L. Martin, DISA/JIEO/Center for Software

High Assurance MLS Database Applications 443
Tom Haigh, Secure Computing Corporation
Dick O'Brien, Secure Computing Corporation
Dan Thomsen, Secure Computing Corporation

Using a Secure Guard with Data Replication 457
Rick Uhrig, Sybase, Inc.

The Impact of Declassifying National Security Information On Data Management 465
Hernan I. Otano, Richard S. Carson & Associates, Inc.

Making the Transition from Data Management to Information Asset Management - Lessons Learned Within the Law Enforcement, Intelligence and Defense Communities 491
Barbara J. Dutton, James Martin Government Intelligence, Inc.

Automating Information Exchange Between Self-Describing Databases 505
Lisa Sills, Georgia Institute of Technology
James Coleman, Georgia Institute of Technology

DAMSL, A Journey Toward The Data Conversion Holy Grail *
Ian Campbell, Wellic Limited

Building the Infrastructure for Client/Server Applications 515
Barbara Timblin, Symantec Corporation

Using Automated Workflow Systems and the Internet to Manage Corporate Data Standardization 527
Bonnie L. McHenry, NCI Information Systems, Inc.
Peter J. Magee, NCI Information Systems, Inc.

Migrating to Open Systems and RDBMS Technology 539
David Thompson, Acucobol, Inc.

Repository Implementation in a Legacy/Reengineering Environment 585
Amy King, Decision Systems Technologies, Inc.
Mark Koltz, Decision Systems Technologies, Inc.
Tanya Jones, Decision Systems Technologies, Inc.
Karen Stanford, Decision Systems Technologies, Inc.

Automatic Data Extraction from Free Text Messages 593
Chris McNeilly, Sterling Software, ITD
Louise Osterholtz, Sterling Software, ITD
Richard Lee, Sterling Software, ITD
Dr. John Hermansen, Language Analysis Systems, Inc.

Creating a Common Analytical Framework Using an Object-Oriented Approach *
Linda Rae Dasher, Richard S. Carson & Associates, Inc.

Linear Scalability on Decision Support Systems: Cray CS6400 603
Brad Carlile, Cray Research, Inc.

The Future of Information Technology *
Tjakko Schuringa, Covia Technoligies Europe

Managing Interface Migration in DoD 613
M. Cassandra Smith, The MITRE Corporation
Susan L. Ficklin, The MITRE Corporation
Darin S. Satterwaite, The MITRE Corporation
David J. Connolly, The MITRE Corporation

Pitfalls, Traps and Gotchas in Building a Large, Multi-Level Secure Database 617
Mike Lefler, PRC Inc.
James Bradley, Coleman Research Corporation

On-Line Processing as a Data Access Method
Maureen Armacost, Richard S. Carson & Associates, Inc.

Lessons Learned in Legacy Data Access 623
Diane C. Reilly, Richard S. Carson & Associates, Inc.
Jeanine A. Fleming, Richard S. Carson & Associates, Inc.

A Practical Introduction to Database Mining *
Dr. H. Stephen Morse, MRJ, Inc.

The Impact of Workflow on Data Management 631
D.D. Marks, Richard S. Carson & Associates, Inc.
J.K. Wass, Richard S. Carson & Associates, Inc.

Index of Authors 643

*Paper not available at time of printing; however, we do expect it to be presented at the Colloquium.

Included
Available
not numbered

APPLICATION OF OBJECT TECHNOLOGY FOR DATA MANAGEMENT

ABSTRACT OF FEATURED ADDRESS

By

Dr. Bhavani Thuraisingham

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730

Data Management is the process of (1) understanding the current and future data needs of an enterprise and (2) making that data optimally available to support the operations of that enterprise. It includes methods for providing integrated access to one or more databases possibly heterogeneous in nature as well as methods for designing and maintaining the databases. Database management is an aspect of data management. Data management has interactions with other technology areas like mass storage management, distributed processing, knowledge management, information management, and human computer interaction. For example, the data extracted from the databases may be presented in an appropriate manner using techniques from human computer interaction.

The key feature of object technology is an underlying model which views the world as a collection of objects which interact with each other through exchanging messages. Encapsulation, where information in an object is accessed through well defined interfaces, and Inheritance, where objects inherit properties from other objects, are two major constructs of an object model. Initially, object technology was applied to develop programming languages. However, during the past decade, there have been numerous applications of object technology in the development of operating systems, data management systems, distributed applications, and heterogeneous systems. Object technology is now increasingly used to design and develop complex software systems.

Object technology is being applied to several aspects of data management. Many database management systems now have object-oriented programming language interfaces. Object-oriented data models are being utilized for database management systems. The resulting systems are object data management systems. These systems include object-oriented database management systems, object relational database management systems, and extended relational database management systems. Object-oriented design and analysis techniques are being applied to design database applications. For example, the entities of the application and their structural properties, the interactions between the entities, and the functions on the entities are being modeling using the object-oriented approach. Object technology is being

applied to handle certain problems in heterogeneous database integration. One application in this area is to utilize an object model as the global data model to represent the metadata information in the databases involved. Another application is the use of distributed object management systems such as the Object Management Group's Common Object Request Broker Architecture to facilitate interoperability between heterogeneous database systems.

This presentation will first provide an overview of data management and object technology. Then it will describe the application of object technology for data management mainly in the following areas: (1) programming language interfaces, (2) data modeling for database management systems, (3) designing database applications, and (4) heterogeneous database integration. Emerging object-oriented standards for data management as well as future directions in the application of object technology for data management will also be presented.

Biography of Dr. Bhavani Thuraisingham:

Dr. Bhavani Thuraisingham is a Principal Engineer with the MITRE Corporation's National Intelligence Division and heads the Corporate Initiative on Data Management Research. She is also a strategic technology area leader in the Advanced Information Systems Center and is responsible for the data and information management section. She is currently working on realtime database systems, massive multimedia database management, and database security. Her interests also include heterogeneous database integration, and object-oriented design and analysis techniques for developing various information systems applications. Her previous work at MITRE included the design and implementation of a secure distributed query processor, database inference controller, and secure multimedia/object-oriented database system. She provides directions in database management research and development for the Department of Defense and the Intelligence Community. She is a Co-Director of MITRE's Database Specialty Group and serves in the Corporate Technology Area Council in Database Systems.

Prior to joining MITRE, she conducted research and development activities at Honeywell Inc. where her work included the design of the secure database system Lock Data Views, the design of a network operating system for space station applications, and also the application of object-oriented technology for developing next generation process control systems and for integrating heterogeneous data dictionaries. Before that she was at Control Data Corporation where she worked on the product development of CDCNET. She was also an adjunct professor of computer science and a member of the graduate faculty at the University of Minnesota.

Her work has been published in over two hundred technical papers and reports including over forty journal articles. She is an inventor of a U.S. patent for MITRE on database inference control . Dr. Thuraisingham gives tutorials in object-oriented database systems, heterogeneous database systems, secure database systems, and realtime database systems to various government organizations, has co-edited a book on secure database systems for North Holland and one on object-oriented systems for Springer, serves on the editorial boards of the Journal of computer security and the Computer standards and interface journal, and has served as the program chair or program committee member at conferences/workshops. She gives invited presentations including the featured address at the 1994 DOD Database Colloquium. Dr. Thuraisingham received the M.S. degree in Computer Science from the University of Minnesota, the M.Sc degree in Mathematical Logic from the University of Bristol, U.K., and the Ph.D. degree in Computability Theory from the University of Wales, Swansea, U.K. She is a member of the ACM, IEEE Computer Society, and the British Computer Society.

Defense Information Infrastructure:
Data Migration Tasks, Techniques, and Solutions

Phillip Cykana
Defense Information Systems Agency
Center for Software

INTRODUCTION

The Defense Information Infrastructure (DII) initiative is a major Department of Defense effort to improve information systems support to the warfighter. Focused on improving systems support in connection with optimizing the use of designated migration systems, this paper is devoted to describing the role of data administration and data management in achieving DII goals.

BACKGROUND

The DoD DII Master Plan (Version 2, 20 March 1995) provides detailed information on the major elements of the DII initiative. These include: DoD mission area applications, tactical applications, communications networks, data standards, value added services, and technology support. As detailed under the Master Plan, the DII is:

> ...a seamless web of communications networks, computers, software, databases, applications, and other capabilities that meets the information processing and transport needs of DoD users in peace and in all crises, conflict, humanitarian support, and wartime roles. It includes:
>
> The physical facilities used to transmit, store, process, and display voice, data and images.
>
> The applications, engineering, and data practices (tools, methods, and processes) to build and maintain the software that allows ... users to access, manipulate, organize, and digest proliferating quantities of information.
>
> The network standards and protocols that facilitate interconnection and interoperability among networks and systems and provide security of the information carried.
>
> The people and assets which provide the integrating design, management and operation of the DII, develop applications and services, construct the facilities, and train others in DII capabilities and use.

In working data administration and data issues under the DII, several points are relevant. First, the DII is about databases. The design, development, implementation, and deployment of data resources that support the range of users in the Department. Second, the DII is about the physical facilities used to transmit and store data. This includes but, is not limited to the use of electronic commerce/electronic data interchange (EC/EDI) technologies such as EC/EDI applications, EC/EDI gateways, Network Entry Points (NEP), and the design and development of EC/EDI databases. Third, the DII is about

data practices (tools, methods, and processes) that are used to build the DII. In the DoD environment this includes tools to support data migration, data modeling, data extraction and load, data quality, and the use and management of data repositories for AIS design and development. Data practices also includes the methods and procedures used to establish and implement DoD data standards as well as the design/development of EC/EDI implementation conventions. Fourth, the DII is about people. The people that work data migration and data integration across the DoD Components. This includes central design activities, megacenter operations, functional data administrators, and database administration personnel.

Generally, data plays an important role in the DII as a mechanism for achieving system integration and interoperability. We have seen this from a number of points of view.

- Operation Desert Storm taught us that the availability of information is key in warfighting. Conrad (1994), for example, cites the lack of intransit visibility of supplies as a major logistics problem. For although our logistics systems were designed to get assets to fixed points of supply, these systems were not able to track and update supply information for units on the move. In this situation, "...thousands of containers filled with undeliverable goods proved to be a vast waste of resources we cannot afford in the future".

- Delays in getting the right data to military personnel can be costly. For example, data delays have been cited by senior DoD officials for lost aircraft over Bosnia under UN sanctioned missions.

Data integration and system interoperability are often major elements in disaster avoidance. In addition, we also have experience in information technology modernization that has shown us that data centered solutions tend to avoid the design/development of stovepiped capabilities. Within the Center for Software, we are using DoD data standards to promote data interoperability under the Global Command and Control System (GCCS) AIRFIELDS project. We are also using data integration and migration techniques to facilitate enterprise integration, design/development of shared database capabilities, and the reduction in the number of AIS that satisfy the same requirements.

DOD MIGRATION EFFORT

Data migration and data integration in the DoD environment has focused on several goals. These have been outlined under public law and mandates issued by the Department. For example, Section 381 of the National Defense Authorization Act for Fiscal Year 1995 requires the Department to establish performance measures and management controls for:

- Accelerating the implementation of DoD migration systems.
- Establishing and using DoD data standards.
- Improving DoD business processes.

Performance measures and management controls are intended to ensure that the Department receives the maximum benefit possible from the development, modernization, operation, and maintenance of DoD migration systems.

In addition to performance measures, recent mandates requiring the selection of migration systems and the acceleration of data standardization and process improvement (DEPSECDEF, 13 Oct. 1993) have underscored the requirement to detail data migration strategies that will help the Department in achieving DoD goals. Major goals include:

- Optimizing the use of migration systems.
- Adopting and using DoD data standards.
- Improving the quality and utility of DoD AIS.
- Reducing AIS operations and maintenance costs.

DOD MIGRATION STRATEGIES

Achieving the DoD goals for migration systems is no easy task. Nevertheless, the DoD community is working to achieve these goals through the use of several types of migration strategies: AIS Consolidation, Middleware, and Reengineering.

AIS Consolidation: This approach is driven by the megacenter migration shown in Figure 1. AIS consolidation is characterized by moving legacy data to the migration environment and then dealing with the interfaces that must be mirrored to support users at particular sites. This approach to optimizing the use of a migration system may be used under situations where selected migration systems can be hosted on the same or similar platform at the megacenter site and/or when terminal or workstation connectivity to an existing or activated megacenter is practical.

The driving force behind the AIS consolidations is the reduction in facilities that is the result of the movement of Information Processing Center (IPC) capabilities to the DoD megacenters.

Figure 1 shows the locations of DoD megacenters and number of facilities being consolidated due to base realignment and closures (BRACs) and Defense Management Review Decisions(DMRDs).

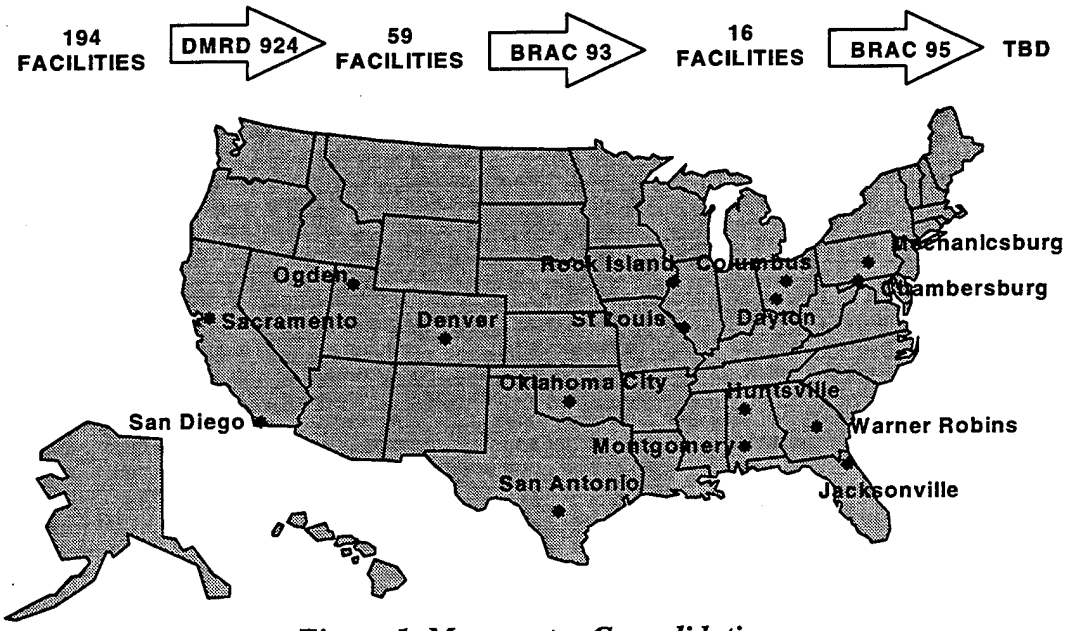


Figure 1 Megacenter Consolidations

In the Department, the first stage of the megacenter consolidation effort is a simple "drag and drop" of processing and data from existing sites to a megacenter. Benefits to be realized by this approach include a standardization of operating environments and naming conventions and improved automation of computer operations and operating support functions. The major factors influencing this approach are cost and availability of modern telecommunications capabilities and other technical considerations. Importantly, the program to consolidate operations does not in itself suggest a requirement to modify the underlying data structures that support migration or legacy systems. Nevertheless, some exceedingly complex data issues arise. For example: (1) legacy systems moved to the megacenter must consider how legacy interfaces will be maintained; (2) data security, data quality (e.g., availability timeliness), data distribution, and remote data entry decisions must also be considered.

Middleware: Generally, the middleware solution is devoted to locating/finding data rather than moving data to migration environments. As such, "middleware" refers to a variety of products and techniques that are used to connect users to data resources.

Figure 2 displays the position of various middleware components as a messaging layer between client and server. It shows legacy systems connected through a function server and an interface while standard or target systems are connected through an object broker.

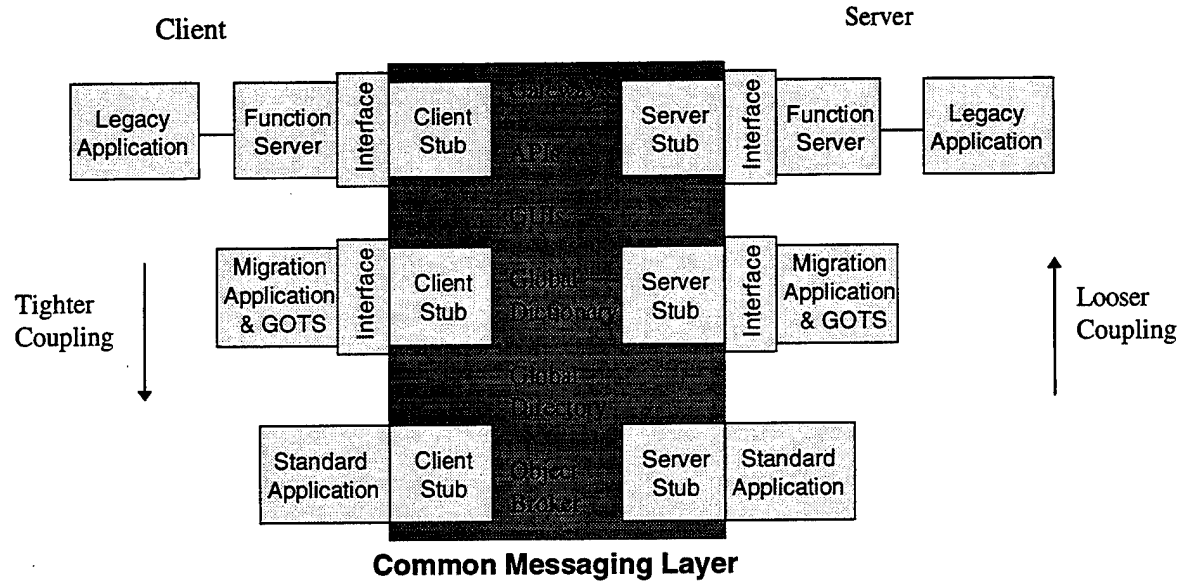


Figure 2 Middleware

Middleware components can be classified by the degree of "coupling" between the user and the data resource. Loosely coupled products allow flexibility in specifying relationships and mappings among data items, but this may also promote multiple semantics and possibly different mappings (e.g. non-standard structures). Tightly coupled products place more authority with standard interfaces and database administrators. Such products support the goal of providing location, replication, and distribution transparency, but demands the identification of authoritative data sources.

Focused on couplings between users and data, it is beneficial to classify middleware solutions as loosely coupled or tightly coupled middleware options. Some characteristics of each option are provided below.

Option 1. Loosely Coupled Middleware: This option is typically used under "quick fix" scenarios and usually allows for the introduction of "False Front Interface" and Graphical User Interface (GUI) technologies at the workstation. Application Program Interfaces (API) are used or developed in conjunction with the use of GUI tools with "point-and-click" icons placed at the desktop. This middleware solution may allow for the performance of X-terminal sessions over communication networks.

This option requires no modifications to the underlying migration/legacy data structures. It can be used to migrate text-based dumb terminal applications to a workstation GUI environment. Via translation code executing on a workstation, text terminal sessions can be captured, and a GUI presented to the user. Keyboard mapping and multiple login problems will need to be resolved. This option can also allow users to transparently access multiple equivalent migration systems via a single standard interface.

Option 2. Tightly Coupled Middleware: This option is the most aggressive middleware strategy. It combines API/GUI technologies and extended data communications with the

design/development of dictionary/directory capabilities that provides data access across a distributed data environment. The dictionary/directory services are typically contained in a global dictionary/directory (reference library) that is located between the workstation and migration/legacy systems. The global dictionary/directory not only locates data/applications but may provide a number of other services (e.g. data replication/distribution, query translation, remote data access, and phased updates).

The aggressive nature of this option requires data standardization and reengineering activities. It should be attempted only after specific objectives have been defined, application and process boundaries have been established, and data standardization has been initiated in parallel with design/development efforts.

The dictionary/directory capability tightly couples users to a three tiered systems environment as shown in Figure 3 This approach may require extensive mappings between the workstation and legacy/migration systems.

The first tier is the workstation environment; the second tier is the global data dictionary/directory database; the third tier is represented by the legacy and migration systems. For the most part, the extensive mappings suggested through the use of a global dictionary/directory are stored at Tier 2. This middleware option promotes interoperability by providing a standard user interface at the workstation, by handling of communications protocols, and through access to data/applications resident on many different hardware, application, or DBMS platforms. The different applications and DBMSs interoperate at the desktop level. Although no modifications are usually made to the underlying migration/legacy data structures (Tier 3), considerable effort may be extended to develop the dictionary/directory capability. Additional information on middle products can be found in DISA *Consumer's Guide to Middleware*, Version 1.0, May 1994

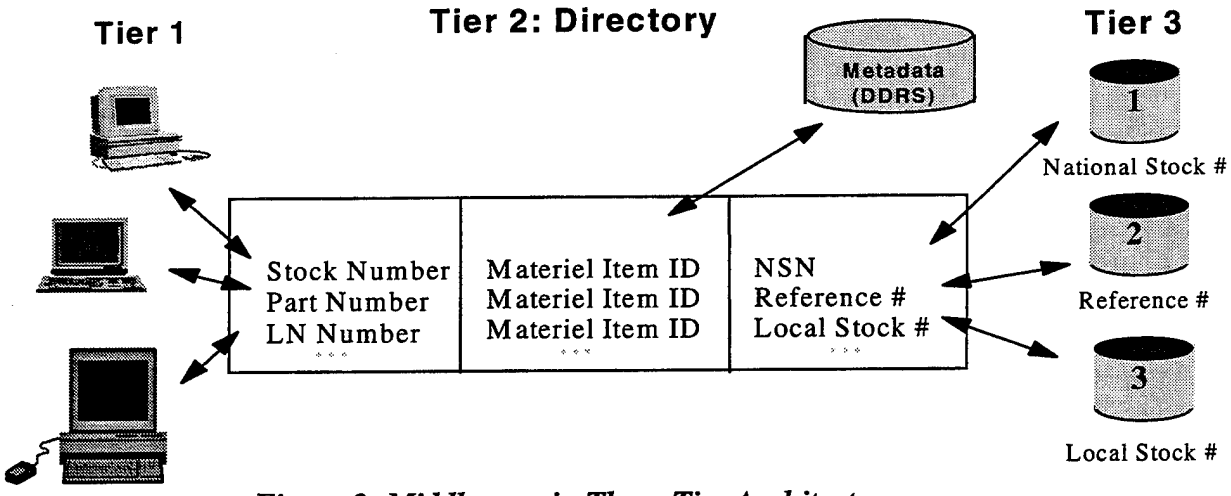


Figure 3: Middleware in Three Tier Architecture

A disciplined approach to the use of middleware may provide benefits that outweigh costs. For example, the utilization of middleware technologies may help to optimize the use of a migration system as the authoritative source of information while maintaining connectivity

to legacy databases for archive or read only purposes. Another approach is to map all legacy and migration data to the tier two directory/dictionary and to eventually move data to the DBMS data structure described in the tier two database. In the Department, the application of middleware solutions may be justified in situations where legacy data must be accessible because of congressional and/or legal requirements. The major factors influencing this approach are: (1) consequences of lost functionality, (2) performance across the distributed environment, and (3) costs connected to the design, development, and implementation of the global data dictionary/directory.

Reengineer: This approach to data migration is devoted to reengineering the migration system(s) to support both common and unique requirements that are supported by migration/legacy systems. As an approach to migration, data and/or application programs may be rebuilt. Importantly, the rehosting of the data is accompanied by software development which is aimed at replacing the existing application programs with either Ada code (FIPS PUB 119) or SQL compliant (FIPS PUB 127) data manipulation language (DML).

The following definitions outline several methods of reengineering that are typically used in conjunction with DoD migration system efforts.

Business Process Reengineering - The fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance, such as cost, quality, service, and speed. (Michael Hammer and James Champy, *Reengineering the Corporation.*) In the DoD environment, process reengineering has been focused on Functional Process Improvement (FPI) initiatives. Although not as radical as the BPR methods (sometimes referred to as "management by carnage"), FPI emphasizes the same fundamental rethinking of business processes.

Information Engineering - A systems development methodology comprising a toolkit of disciplined techniques supported by automated tools within a clear project management task structure. IE starts with three architectures, (Information, Business Systems, and Technical), which are detailed until code generation can be done. This is the basis for most integrated CASE tools.

Systems Engineering - Modifications and enhancements to existing migration or legacy systems, which can include expansion of functionality, interface enhancement, forward or reverse engineering, technology refreshment, or architectural reconfiguration (such as to client/server). It focuses on increasing the value of the migration systems in the legacy inventory.

The three reengineering methods can also classified as pertaining to two distinct reengineering options: (1) incremental replacement/evolutionary reengineering and (2) "big-bang" design, development, and deployment. Either option can be used under a migration effort.

Some examples of incremental and evolutionary changes that may be made to migration systems include:

- Conversion from one file format to another and the rehostingof the application under the same or a different hardware/software platform and operating system.
- Data structure is salvaged; the code is reworked to support integration and interoperability.
- Transition from flat file data structures to a DBMS and salvage of the existing procedural language by inserting database access routines into the existing code.
- Both the data structure and application code are reworked to support enterprise information integration and interoperability.
- Database consolidation

Typically, big-bang reengineering attempts to leap past the optimized use of migration systems and focuses on the design/development of "target" environments. Centered on "targets" or the "objective system", the BPR method may be used to support a fundamental restructuring of the organization and the automated support provided across functional areas. FPI and information engineering efforts may be structured to support the same purposes as BPR. As "big-bang" approaches to migration, FPI and information engineering efforts can be devoted to describing the "target" environment through extensive "as-is" and "to-be" modeling.

Some examples of "big-bang" reengineering efforts include::

- Large cross functional AIS design/development efforts that involve extensive changes to the business rules governing the creation, management, and use of data.
- Large AIS replacement efforts with new requirements.
- Reengineering efforts that throw it all out and ignore the past.
- New Starts: Top-Down approaches that are devoted to strategic planning, business area analysis, system design, AIS development, and finally deployment.

DATA MIGRATION TECHNIQUES

In conjunction with the execution of the strategies outlined above, several major tasks, techniques and technologies are being used to support the movement of DoD users to selected migration systems.

Five of the major data migration techniques and tasks that are being used across the Department include:

- Reverse and forward engineering.
- Implementing DoD data standards.
- Managing interfaces.
- Performance of data extraction and load.
- Conducting data quality projects.

Given the complexity of many of the Department's systems and mandated goals to optimize the use of migration systems, DoD data administration and database administration presents special problems, issues, and opportunities. Central to these concerns is system redundancy and the techniques and tasks to move legacy systems to the migration environment. As shown above, there are a number of strategies that can be used to "move" legacy data. Our experience shows that some functional areas prefer some techniques over others. For example, functional areas with prior investments in system modernization are likely to choose reengineering options. In managing change from legacy to migration systems, these functional areas are using reverse engineering and forward engineering to: (1) document legacy requirements (reverse engineering) and (2) enhance the capabilities of the migration system (forward engineering).

Functional areas concerned with the proliferation of data redundancy have focused efforts to establish and implement DoD data standards. In this process, they are working the creation and approval of data standards under DoD 8320.1 series guidelines and are registering their use of DoD data standards. Working with these functional areas, the Center for Software has worked criteria that is used to determine whether migration system application elements either match or should be mapped to DoD data standards. Criteria include metadata characteristics such as: data type, field length, domain values, and access name. Our initial matches and mappings using the GCCS AIRFIELDS project shows approximately 70% against DoD standards.

Another data migration task that has received much attention in the DoD community is interface management. In situations where the migration system is to support various types of interfaces, integration managers are using a number of techniques to manage data interchange. For example, interface management techniques may include the use of interface databases, data warehousing, data translation, and middleware solutions. In managing interfaces, some integration managers are working the design/development of interface architectures. These architectures recognize the need to match interface requirements (e.g., read/write access, batch data load, on line transaction processing (OLTP), decision support) to appropriate interface solutions (e.g., data replication, point-to-point data interchange, data warehousing).

Working with the logistics community, the Center for Software has also worked data extraction and load problems under the Depot Maintenance Management Information System (DMMIS). Under this project, data is being extracted from legacy systems and loaded to an interim database that is used to support data quality checks against the data prior to loading to the DMMIS. In combining data extraction and data quality tools this project demonstrates the feasibility of cost effective measures that can speed the load of

migration systems from legacy data sources. It also shows how integration managers can incrementally develop a data integration and migration environment.

Integration managers dealing with data migration are working with some tough problems. To solve these problems, they are working in environments that must support a range of migration objectives. These include: (1) movement and improvement of data, (2) shut-down of legacy systems, (3) data engineering and reconciliation (e.g., identification and description of common and unique data needs), (4) design, development, and deployment of migration system enhancements, (5) management of data interchange and (6) design, development, and deployment of middleware solutions.

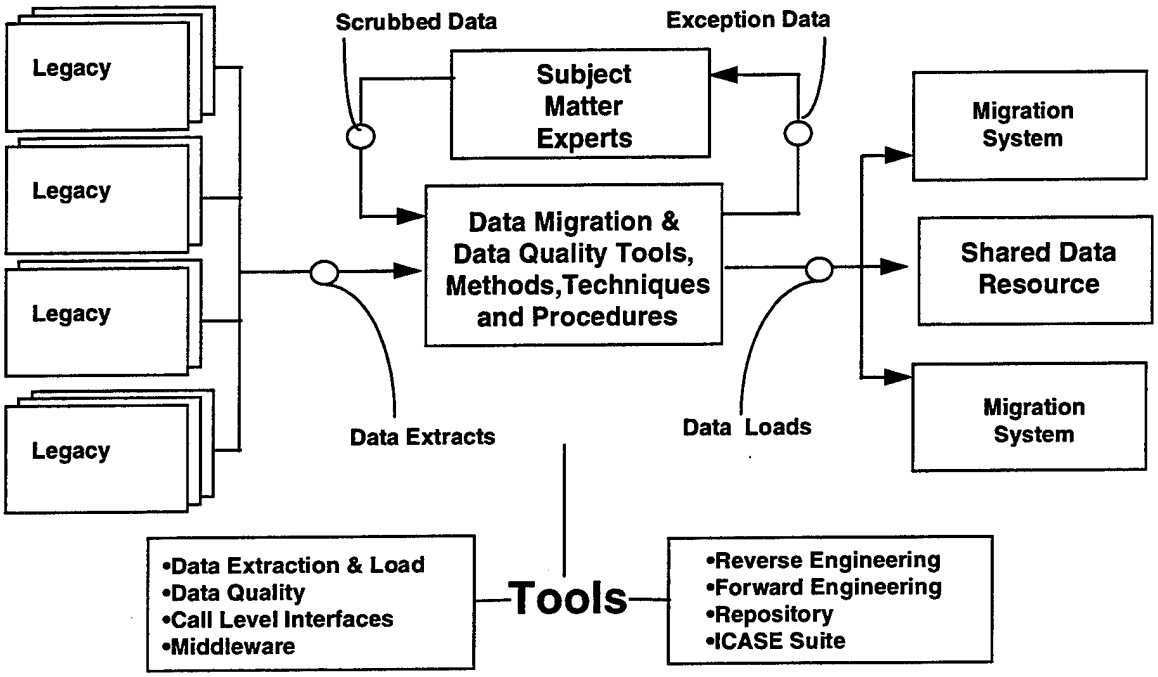


Figure 4: Data Integration and Migration Environment

Importantly, integration managers and Central Design Activities (CDA) have a wide range of tools, techniques, and methods that can be put to use to achieve the desired results. As shown in Figure 4, the development and use of the data integration and migration environment can ease data migration work. First, it contains all the tools (e.g., data quality, data migration, repository, and design/development) required to execute data engineering and migration tasks. Second, it includes technical personnel and subject matter experts that are teamed to work operational data integration and migration issues.

CONCLUSION

This paper has been devoted to describing the importance of data administration under the DII and common migration strategies that are used across the Department. In conjunction with these strategies, this paper has also detailed several major data migration tasks, techniques, and solutions that are being used within the Department to support data sharing, data integration, system interoperability, and the management of data

quality/utility. As the Department moves to shared data resources, the disciplined execution of both data integration and migration tasks will demand increased attention. Tools, techniques, methods, procedures, and knowledgeable personnel will be indispensable as designated migration systems evolve toward target environments.

REFERENCES

Conrad, Scott W. Moving the Force: Desert Strom and Beyond, National Defense University, December 1994.

DISA Consumer's Guide to Middleware, Version 1.0, May, 1994.

DEPSECDEF (13 Oct. 1993) Office of the Deputy Secretary of Defense Memorandum, " *Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement," October 13, 1993.*

Defense Information Infrastructure Master Plan, Version 2.0, 20 March 1995.

DoD 8320.1-M, DoD Data Administration Procedures, March 1994.

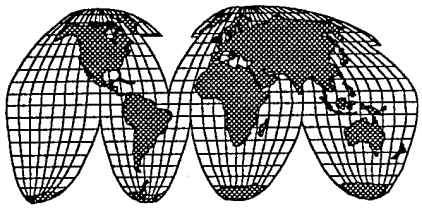
DoD 8320.1-M-1, DoD Data Element Standardization Procedures, January 13, 1993.

DoD 8320.1-M-x, DoD Data Model Development, Maintenance, and Approval Procedures, May 23, 1993.

Hammer, Michael and Champy, James. Reengineering the Corporation, Harper Collins, 1993.

Author Biography

Mr. Phillip Cykana is with the Defense Information Systems Agency/Joint Interoperability and Engineering Organization/Center For Software ((DISA/JIEO/CFSW). He is currently responsible for managing and directing work in the Data Design and Migration Division under the Software Environments Department. Mr Cykana has extensive experience in data administration, information engineering, DoD data standardization, and AIS requirements analysis, design, and development. Mr. Cykana received his B.A. from the University of Wisconsin; M.A. from the University of Minnesota; and M.B.A. from Wright-State University. Recent experience includes data administration work under the Defense Information Infrastructure (DII) initiative; adoption of external data standards under the DoD data standardization initiative; development of the Command and Control (C2) Core Data Model; and data migration work for the Global Command and Control System (GCCS), Standard Procurement System (SPS), Depot Maintenance Standard System (DMSS) and the Materiel Management Standard System (MMSS). Prior to joining DISA, Mr Cykana provided support to the Air Force Materiel Command under the Joint Continuous Acquisition and Logistics Support (JCALS) program, the Joint Uniform Services Technical Information System (JUSTIS), and other logistics automation efforts.

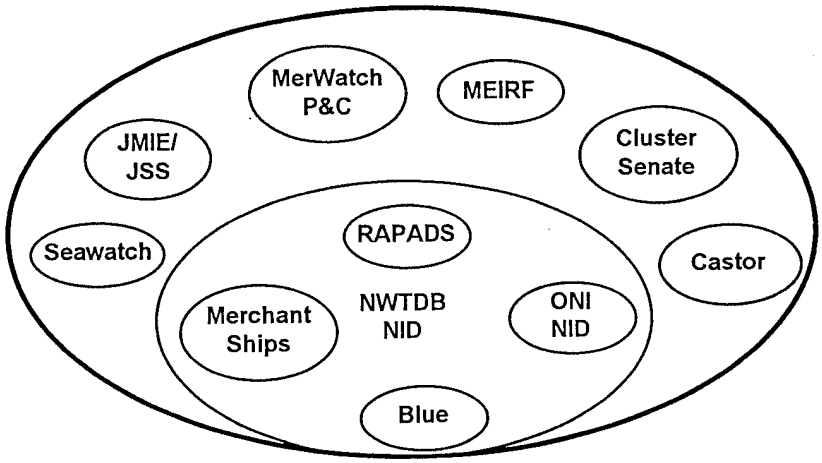


Joyce A. Wineland
ONI-712
(301)669-5413 DSN 294-5413
(301)669-5401 DSN 294-5401
WINELAJ@NMIC-MAIL.DODIIS

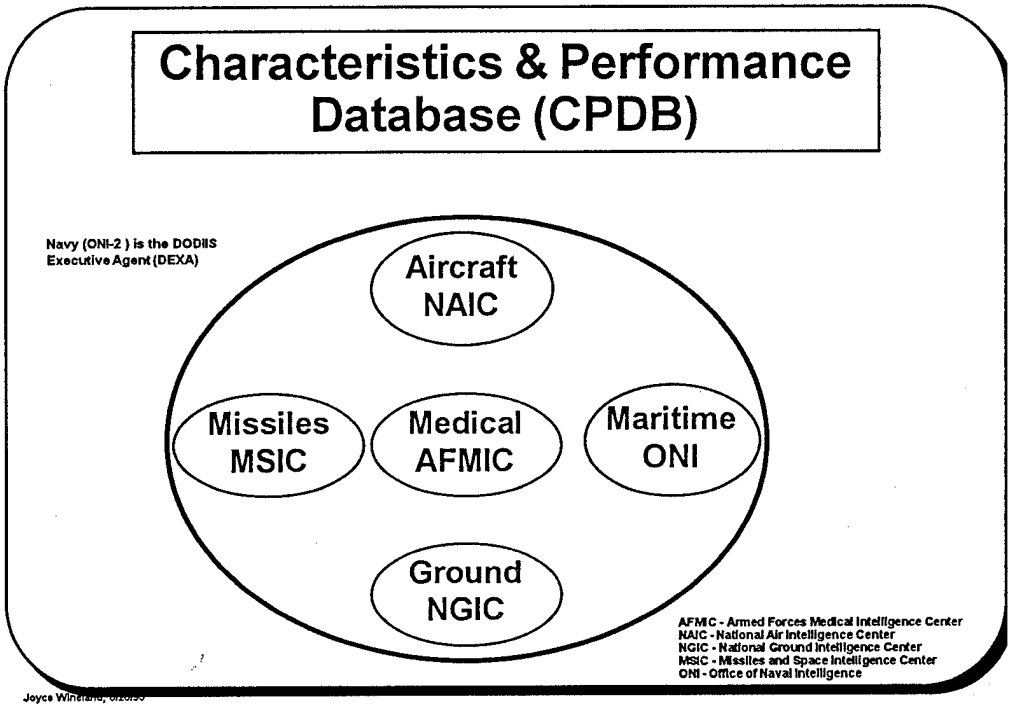
Use of Data Modeling to Coordinate Three Separate but Related Migration Systems

28 August 1995

National Maritime Intelligence Database (NMID)

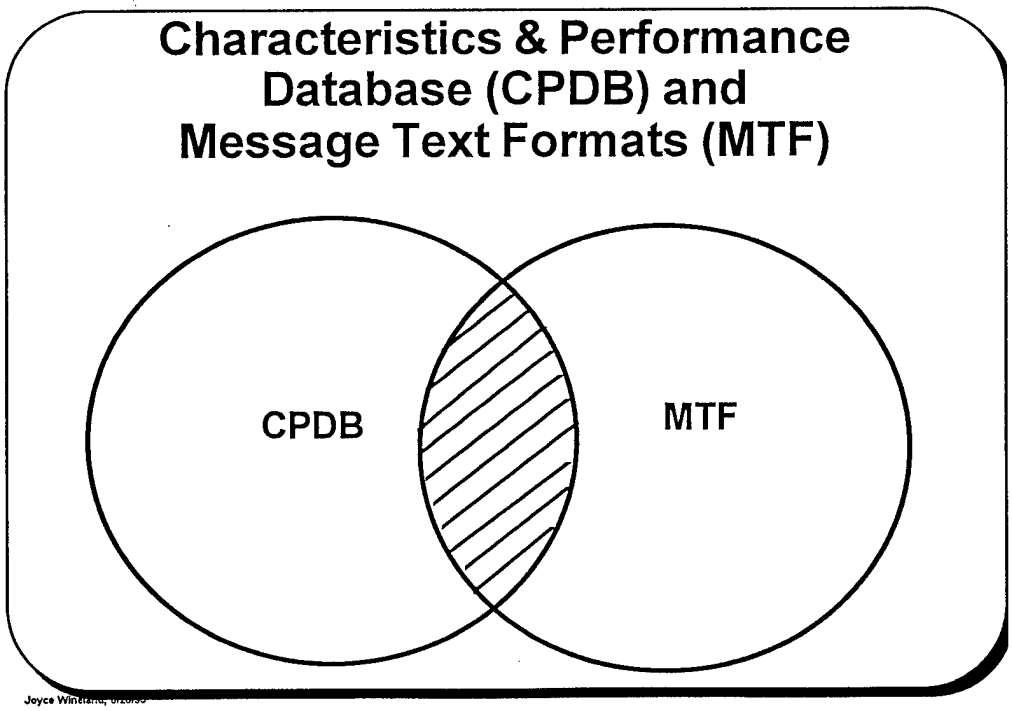


Data Modeling



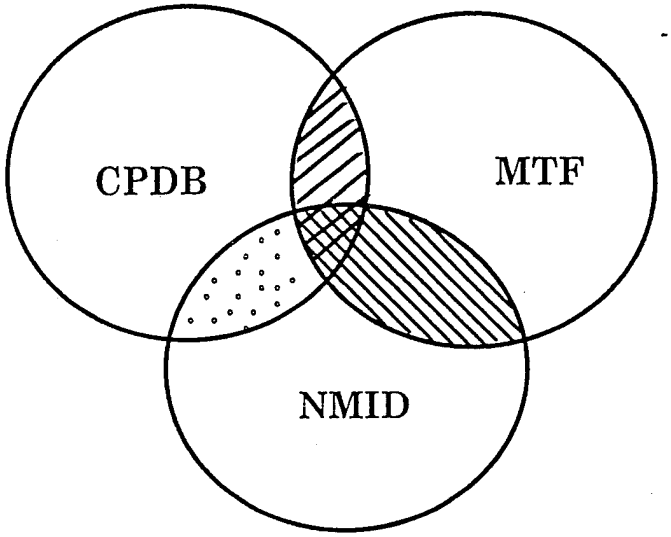
Joyce Winstanley, 6/20/96

Data Modeling



Joyce Winstanley, 6/20/96

Relationship of CPDB, MTF and NMID Data Elements

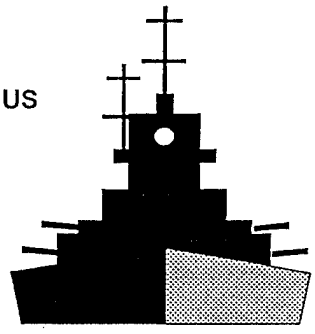


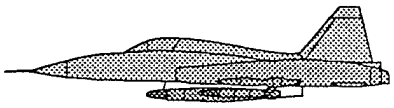
Joyce Wineland, 8/25/95

SITUATION:

Aircraft Detected

US





THEM

Joyce Wineland, ONT-712, 8/25/95

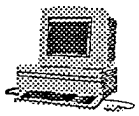
QUESTIONS

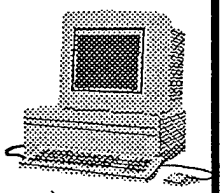
1. What is it? Aircraft type.
2. Who is it? Country, Unit.
3. Where is it? Latitude/Longitude, Range, Bearing.
4. Where did it come from? Airfield or Carrier.
5. What can it do to us? Weapons.
6. How can it detect us? Sensors.
7. Why is it here? Intent.
8. How can we avoid detection? ECM.
9. If detected, how can we defend ourselves? My own weapons.
10. If detected, how may we defend ourselves? Guidance.

•They are asking the same questions.

Joyce Wineland, ONI-712, 8/25/95

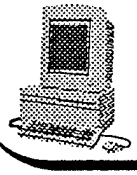
INFORMATION AVAILABLE

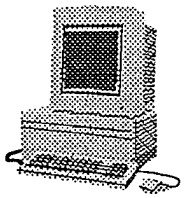


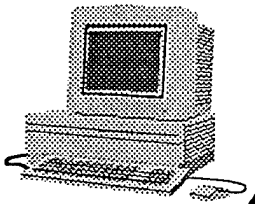


Question-Answer Available

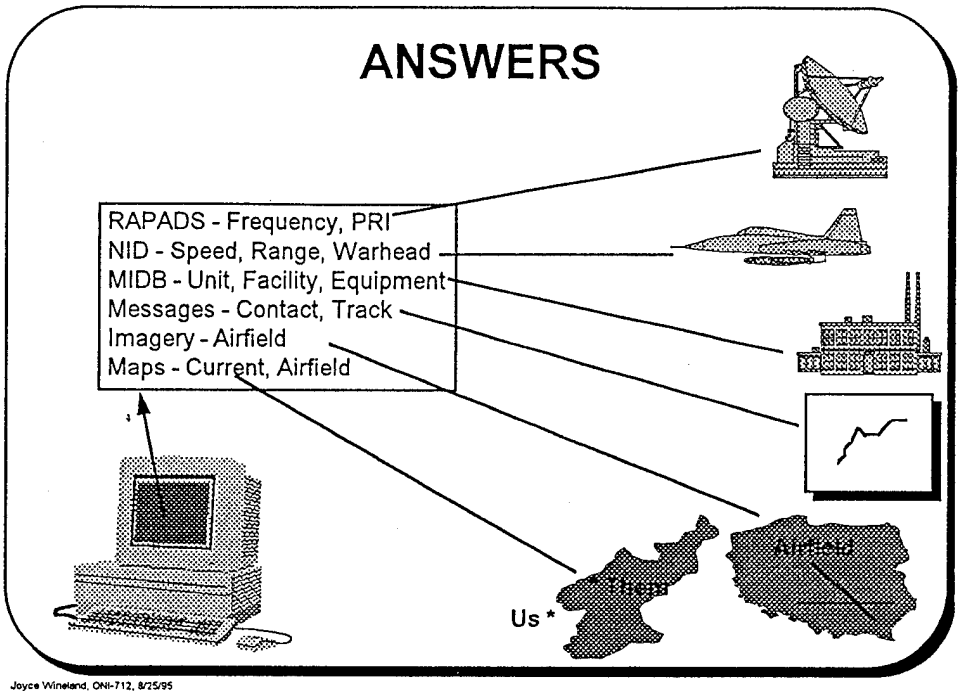
7	Human Knowledge
10	National, Theater, JTF - Guidance
8,9	EWOPFAC - RAPADS (Radar parametrics)
1, 5, 6, 8, 9	S&T Centers - CPDB (Platforms, Weapons, Sensors)
2	DIA - MIDB (Facilities, Units, Equipment)
2	DMA - AAFIF (Airfields)
3,4	Messages - Real-time Information fromOrganic Sensors
4	NRO - Imagery
3	DMA - Maps







Joyce Wineland, ONI-712, 8/25/95



Joyce Wineland, ONI-712, 8/25/95

Data Modeling

BENEFITS

- Message Text Formats can be used to exchange data between databases
- Seamless data exchange, no translators
- Remove the man/woman from the loop
- Faster, more accurate response, from source thru analysis to shooter.

Joyce Wineland, 8/25/95

ISSUES

- Complex architecture
- Data must be independent of system implementation
- Difference Functional Data Administrators (FDAd) for Intelligence and Communications
- Coordinate with all services and several DOD agencies
- Which chain of command for submitting data element packages?
- How to determine which should also be MTF changes vs translator?
  - NATO
  - Time lag for implementation
  - Bandwidth

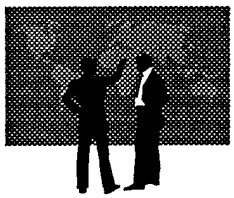
CONCLUSION

Data Modeling is one technique for identifying the data available, normalizing the database structures, removing redundancy, resolving ambiguity, and documenting consistent database linkages.

It is not quick and easy.



Without it, we put our people and nation in jeopardy



DATA SHARING, INTEROPERABILITY, AND STANDARDIZATION

by Janet E. McDonald
Electronic Proving Ground

An exciting and beneficial methodology has been developed within the Department of Defense (DoD) which facilitates the acquisition, effective use, and sharing of data across organizational and functional boundaries. This paper highlights the essential features of the Joint Data Base Elements for Modeling and Simulation (JDBE) project that developed this new approach to data sharing, interoperability, and standardization through information modeling.

Several years ago, the DoD began putting emphasis on information standardization with the initiation of the Corporate Information Management (CIM) program. CIM has created many DoD policies, directives, and programs which are designed to standardize information processing systems and data. It is now recognized that data precision is crucial, particularly in modeling and simulation (M&S) in support of Test and Evaluation (T&E), and that standardized and accurate representation of data is a top priority in the DoD M&S Master Plan.

Over two and one-half years ago, the U.S. Army Electronic Proving Ground (EPG) recognized the investment in our legacy data bases as well as the need to comply with the emerging DoD data standards. A proposal was submitted to the Defense Modeling and Simulation Office (DMSO). Subsequently, a proof of concept was successfully executed by the JDBE project at EPG. For the proof of concept, we chose the subject area of electromagnetic equipment characteristics and integrated data elements from nine different data bases, some of which included IEW parameters. The JDBE project has now developed *reverse-engineering* and *data integration* methodologies. By applying the reverse engineering methodology, *data models*, describing existing data bases, are created. By grouping data elements by subject area and applying a data integration methodology, mappings can be defined to share data among diverse data sources and user information systems.

Figure 1 depicts why it is worth the effort to develop a data standard, especially when so much has already been invested in the many data bases that are in constant use. Here, five separate and incompatible data base systems need to share data. If no common standard exists, a transformation must be set up for each direction of transfer between each pair of data bases. But, with a single standard, only two transformations (one for import and one for export) are needed for each data base. Even with just five data bases, 20 transformations are required without a standard, but only 10 are needed if a standard exists. This advantage grows rapidly as more data bases are considered; e.g., if 100 data bases wanted to share data, 9,900 transforms would be needed without a standard, versus only 200 with a standard. Notice, too, that this concept allows existing data base systems to continue to function *without alteration*. The capability for data sharing, gained through a bottom-up development of data standards, can be very bene-

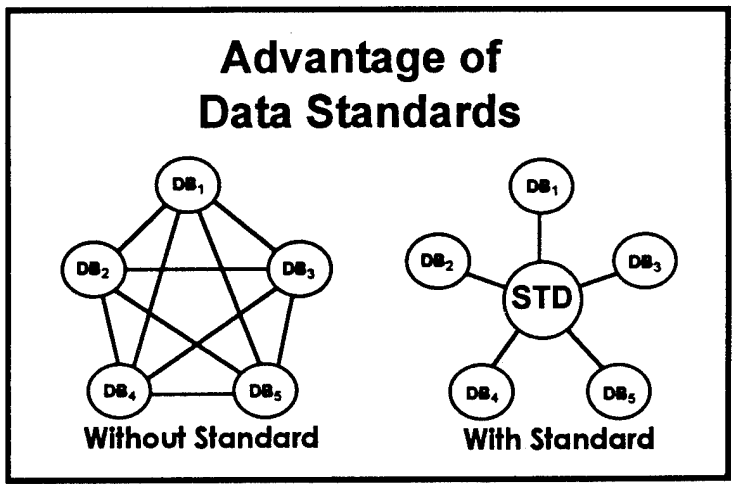


FIGURE 1. Advantage of sharing data through a standard

ficial to many different functions. One of the biggest payoffs of this approach lies in its application to T&E activities.

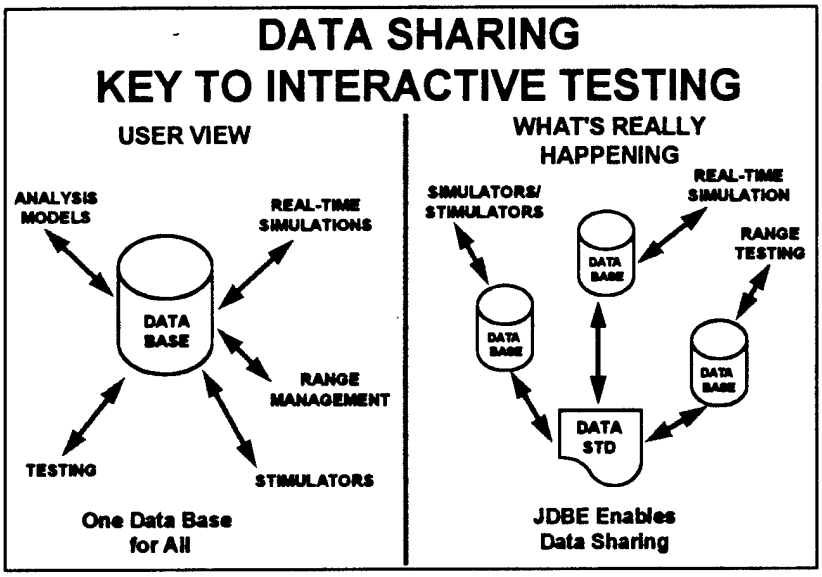


FIGURE 2. Data sharing for Test and Evaluation

So, what can a data sharing capability do for a T&E operation? As shown in Figure 2, the advantage is realized in the user's view of the data base system supporting various T&E functions. The goal is to make all user systems (such as range management, test item stimulation and control, and analytical mechanisms) view and access the data base as a single, integrated system with information freely flowing among them as they interact in the T&E process. In order to make this possible, an integrated data standard is defined which becomes the basis to map data elements among the physically and logically independent data bases that already serve different functions. What is really happening, is that data are being transformed and exchanged as necessary to support those interacting functions. The JDBE methodology may be applied to achieve this data sharing.

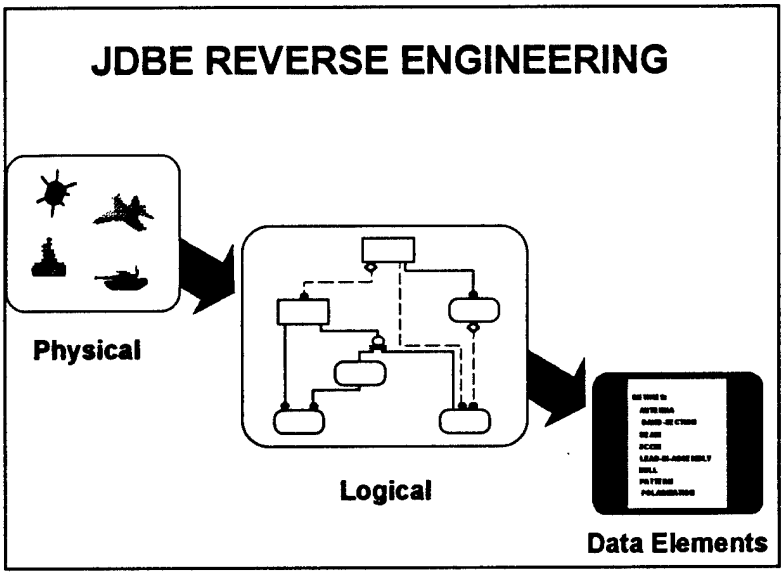


FIGURE 3. Reverse-engineering process

The discussion so far has mentioned using existing data bases, systems that represent major investments and are supporting critical activities on an ongoing basis. Rather than waiting for data standards to be imposed from the top down, and possibly suffering additional expense and an interruption of essential functions, the JDBE methodology proposes standard data elements through a reverse-engineering process. As shown in Figure 3, JDBE captures the logical structure of existing data bases by creating models, expressed in the graphical IDEF1X data modeling language. This language is the subject of a Federal Information Processing Standard (FIPS 184). Groups of such *project information models* are then integrated to define the data standard and the corresponding data transforms to make data sharing possible.

If IDEF1X project information models are developed for data source and user systems which must interact and share data, transforms may be defined to allow the integrated operation of the systems via a central data standard. Figure 4 is an example of data integration in the T&E environment. In this example, data sources 1 through N interoperate with T&E user systems 1 through N via appropriate data transforms.

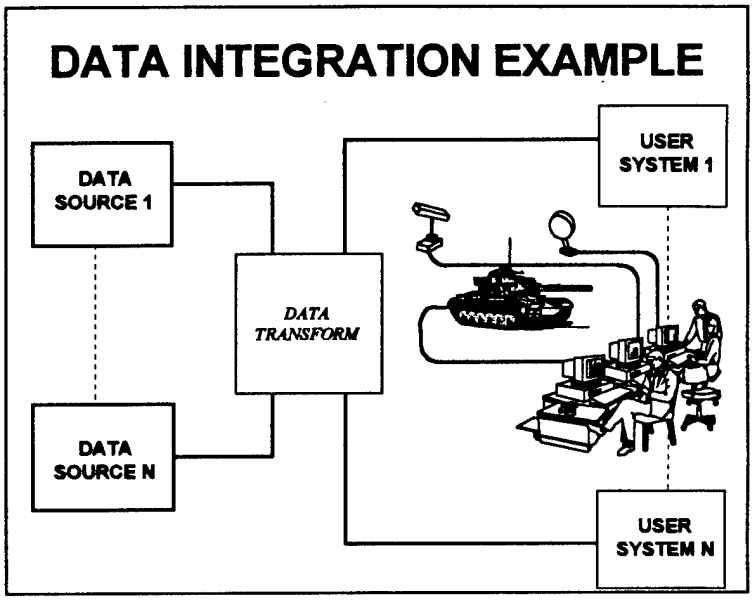


FIGURE 4. Data integration for Test and Evaluation

Figure 5 shows a more general example of how the JDBE methodology of data integration might be applied to a requirement for an integrated, standard data base. Multiple existing data bases, which are independent and non-standard, may provide data that are eventually needed in various user systems. The *capabilities* of these source data bases, and the *requirements* of multiple, dissimilar user information systems (e.g., training or analytical simulators, weapon systems, system testbeds, or decision support systems) are derived through data modeling and are merged into an integrated data model. This model is like a **blueprint** for the creation of an integrated standard data base. Data modeling yields mappings which are the specifications for the

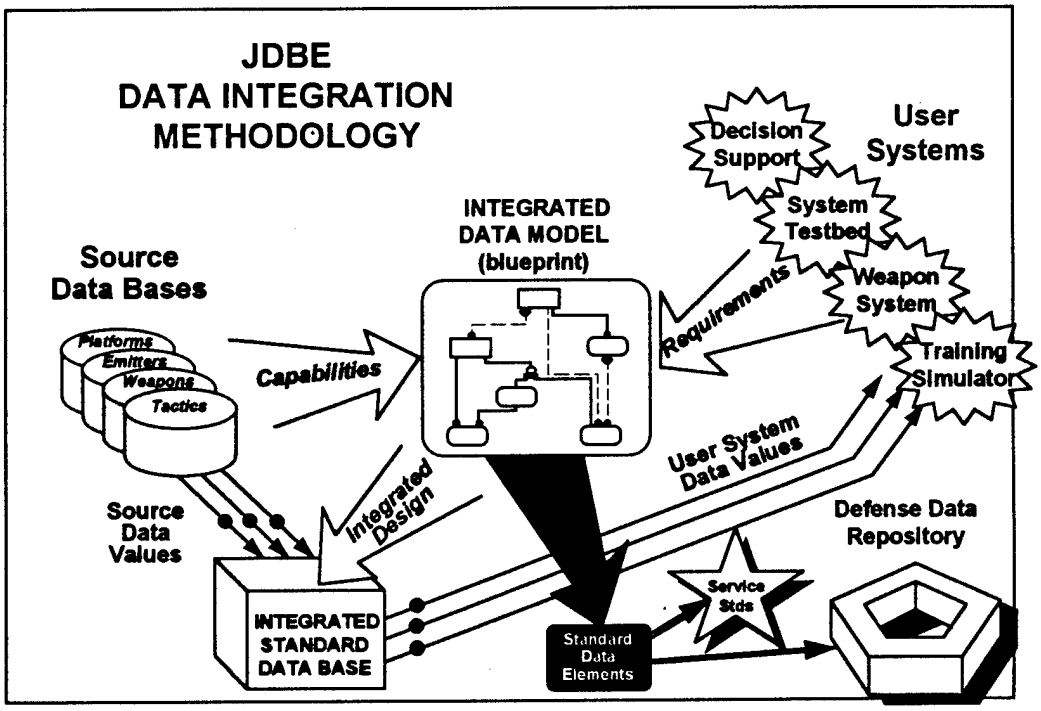


FIGURE 5. General data integration example

transformation of data between data sources and user systems. This methodology is currently being applied by the JDBE team to design a common threat data base for a joint project, the Universal Threat System for Simulators (UTSS). The data model also provides the meta-data (data about data) needed in *Army or other service data standards* and for submission of *candidate standard data elements* for inclusion in the Defense Data Dictionary System (DDDS).

The main thrust of the JDBE methodology is to support **data sharing** within the defense community. Currently, the JDBE project is attempting to aid appropriate DoD organizations in realizing this goal. The methodologies and procedures for their application are documented in the **JDBE Methodology Manual**. This manual is available from EPG upon request. The JDBE team has also developed a curriculum and teaches classes on data modeling and the JDBE methodologies.

With completion of the JDBE proof of concept phase, the project has moved into application of the methodology and techniques both in conjunction with other DMSO sponsored efforts, and in support of other client projects. Several of the tasks for DMSO directly support the objectives of the DoD M&S Master Plan.

One of these tasks is the hosting and development of the Interim M&S Resource Repository (iMSRR). The iMSRR is implemented as a collection of World-Wide-Web (WWW) nodes with various resources available at each node. The iMSRR features user authentication and access control to sensitive data while maintaining public front-ends. The WWW architecture permits multimedia interfaces to information and very user friendly interfaces. The JDBE Node offers a repository of data models which are available for both interactive browsing and downloading. The iMSRR hosts directories and catalogs of its own as well as containing links to other popular catalogs and it includes a variety of utilities and tools for Web browsing and editing, data administration, and other reusable software components. The iMSRR also provides a means for distributing information about the various DMSO working groups. It is operational at the Universal Record Locator: http://huachuca-jdbe.army.mil/. The JDBE team is working to expand the repository services and components, to evaluate and develop tools, to contribute to the definition of standards, and to add features to fit emerging requirements. Utilizing what we have learned and are learning in establishing this repository, a similar repository could be set up for use by the T&E or IEW communities.

JDBE methods were first set down to solve data sharing problems in the M&S community, but the processes that were defined have wide applicability to other areas. T&E involves many applications for shared data. There is a growing requirement to use test data to create more realistic "synthetic environments" for training exercises and subsequent testing of equipment, systems, and employment concepts. A central idea in the application of simulation technology is to "train as we fight." This depends on effective use of data from multiple sources, consistently shared, to create appropriate "virtual" testing and training environments and scenarios.

The essence of the JDBE methodology is the attainment of the goal of **data sharing and reuse** within DoD. Although JDBE takes a bottom-up approach, it fully **supports the DoD's long-term standards objectives** and processes. In the near term, **JDBE can assist organizations** in the integration of diverse data resources or in the shared use of common data bases to supply data for multiple applications. The JDBE team includes personnel trained in proven methods and experienced in the techniques to allow sharing of data from several sources. The JDBE methodology facilitates moving data from existing data bases into multiple applications. JDBE directly supports the DoD Functional Data Administrator (FDAd) for M&S and the DoD data standardization and data administration directives.

DEFENSE MODELING AND SIMULATION

DATA ADMINISTRATION

by

Dr. Chien Huo
Defense Modeling & Simulation Office

Ms. Iris Kameny
The RAND Corporation

1. BACKGROUND

The Military Services, Joint Chiefs of Staff (JCS), Office of the Secretary of Defense (OSD), Combatant Commanders, and DoD Agencies are major users of modeling and simulation (M&S) for live exercises and for virtual and constructive simulations. Simulation environments span geographic regions; air, land, and naval forces; and command echelons, and are used in stand-alone or networked modes. The Chairman of the Joint Chiefs of Staff, General John Shalikashvili, has stated that the Services have not yet tapped the potential of using simulation. Given this vast expanse of applications and an acknowledged need for simulation, data quality and shareability are of paramount importance to users of M&S.

Within OSD, the Defense Modeling and Simulation Office (DMSO) was established to promote cooperation among DoD components to maximize efficiency and effectiveness by serving as a full-time focal point for information concerning DoD modeling and simulation (M&S) activities and by promulgating M&S policy, initiatives, and guidance. The Defense M&S Initiative (Reference 1) encourages information sharing, investments in common technologies, and the formulation of common standards for simulation development and interoperability across the training, analysis, and acquisition functional areas in the Components. The May 1991 DoD Executive Council for Modeling and Simulation (EXCIMS) vision statement (Reference 1) alludes to the need for standards that will drive data and database interoperability:

> "Defense modeling and simulation will provide available, operationally valid environments for use by DoD Components ... from *affordable, reusable components interoperating through an open systems architecture."*

Since October 1993, senior managers throughout DoD (References 2, 3, 4, and 5) provided guidance and generic evaluation criteria to be used in the selection of migration systems. One of the critical factors is data.

To accomplish this, DMSO seeks to foster development of DoD-wide standards, databases, and communications capabilities. In this way DMSO intends to promote simulation system interoperability, the development and use of standards-based databases, models, and simulations.

2. NEEDS

Users need to be able to quickly access and acquire operationally valid data in order for M&S environments and scenarios to represent the real world in sufficient detail and resolution to train forces, develop doctrine and tactics, plan and assess operations, perform "what if" analyses of

operational alternatives to potential deployment scenarios, and support the acquisition of new or modified war fighting capabilities. User need for data in the M&S community can be described in the following four phases:

1. locating, accessing, acquiring, and preparing data inputs for M&S,
2. executing the simulation and exchanging data interoperably with other simulations,
3. post-processing of M&S results, and
4. management support for designing model runs and experiments, and maintaining records of experiments.

Data standards and recognized authoritative data sources are required to meet the need, to ensure data and algorithm consistency, especially for interoperating across models. In order to produce valid and useful M&S results, the data used in a model must be verified, validated and certified by the data producer and the exercise or study director as part of the verification, validation, and accreditation (VV&A) process for the model or exercise.

In view of the importance of pre-processing, the M&S community has recently developed a number of data centers to collect, maintain, verify and validate, and provide data either to specific models or to users of models. These include the Army TRAC Automated Data System (TADS), CENTCOM's Conventional Force Data Base (CFDB), the Joint Staff/J8's Operations Analysis and Simulation Interface System (OASIS), the Navy's Automated Repository for Modeling and Simulation (ARMS), and others. The trend is toward developing more data centers, which raises a need for coordination among the centers to promote reuse and sharing of data. Some of these centers are developing data models and standards in order to improve the quality and availability of their data.

3. M&S DATA ADMINISTRATION PROGRAM

The DoD M&S Master Plan (MSMP) (Reference 6) provides a comprehensive framework for planning, programming and budgeting of M&S projects, programs, and activities; and assigns responsibilities for implementation. The MSMP begins with the EXCIMS Vision for M&S and describes how M&S can substantially improve capabilities in each of the four pillars of military capability: readiness, modernization, force structure, and sustainability. The MSMP then presents six objectives (Figure 1) necessary to achieve the M&S vision and examines a baseline assessment to identify shortfalls that must be corrected to realize the vision.

These six MSMP objectives are constructed from three main concepts: provision of a technical framework for M&S; provision for authoritative representations of natural environment, systems, and human behavior; and establishment of an M&S infrastructure.

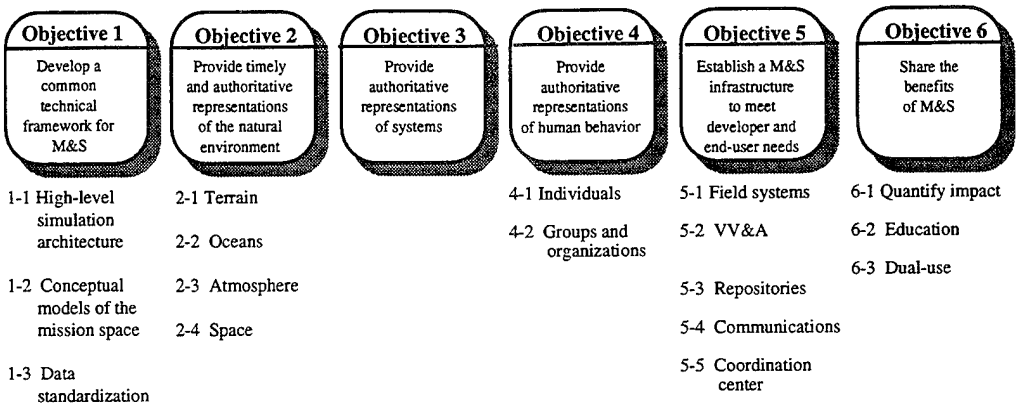


Figure 1. DoD M&S Objectives and Sub-Objectives

DMSO is delegated with the mission and full authority to act as the M&S Functional Data Administrator (FDAd) by the Director, Defense Research and Engineering (DDR&E). It is developing and implementing the M&S Data Administration (DA) program in accordance with the DoD MSMP, DoDD 5000.59 (Reference 7), and DoDD 8320.1-M (Reference 8) to manage these data resources effectively in the M&S community. The mission of the M&S DA Program is to enable data suppliers to provide the M&S community with cost-effective, timely, and certified data to promote reuse and sharing of data, interoperability of models and simulations, and improved credibility of modeling and simulation results. The strategic objectives of the M&S DA Program are to:

- Establish, promulgate, and oversee policies, procedures and methodologies for M&S data requirements; data standards; data verification, validation, and certification (VV&C); and data security to provide quality data as common representations of the natural environment, systems, and human behavior.
- As part of the future DoD Repository (DoDR) system, develop a distributed resource repository system to serve the community in accessing and retrieving M&S resources (metadata, data, algorithms, models, simulations and tools).

The M&S FDAd, together with the M&S Data & Repositories Technology Working Group (DRTWG) and the M&S community, has put in place an infrastructure and is executing the MSMP. The M&S FDAd is coordinating efforts with the Components (Reference 7) to help achieve an M&S framework by concentrating on data necessary for representations of the natural environment, systems, and human behavior in support of C3I applications. These areas support such critical projects as the Global Command and Control System Leading Edge Environment (GCCS LEE) and Synthetic Theater of War-97 (STOW-97).

The M&S FDAd directs data modeling projects to capture data requirements and develop standard data elements, works with data users and data producers to standardize data, ensures the on-going assessment and improvement of data quality, and supports data security. A distributed resource repository system will serve as the primary infrastructure for coordination and will store these M&S resources for standardization and reuse.

By the year 2000, the M&S DA infrastructure will provide Components with distributed repositories that the M&S community can use to access resources for reuse in models and simulations. The M&S DA Road Map in Figure 2 identifies the major activities that will be accomplished over the next several years to provide the data administration products needed across the community to achieve M&S objectives.

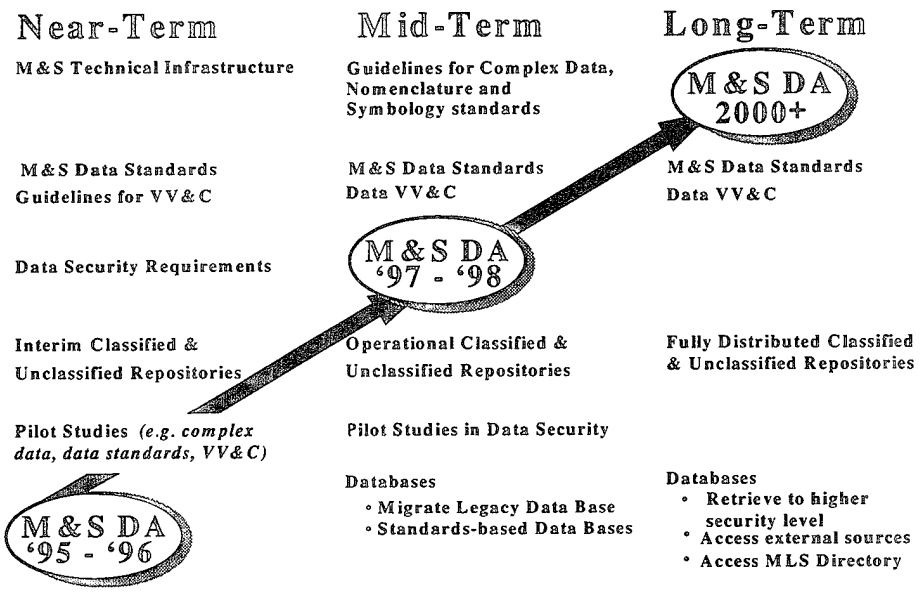


Figure 2. M&S DA Road Map

3.1 Data and Repositories Technology Working Group (DRTWG)

The M&S FDAd has established the DRTWG and its task groups and subgroups as shown in Figure 3. The DRTWG is the focal point of the M&S DA program. It is co-chaired by the authors, who coordinate all activities through the technical infrastructure support groups working to achieve the DoD MSMP objectives. The figure also depicts the M&S DA technical infrastructure, the DRTWG, and their relationship to other Components' DA organizations.

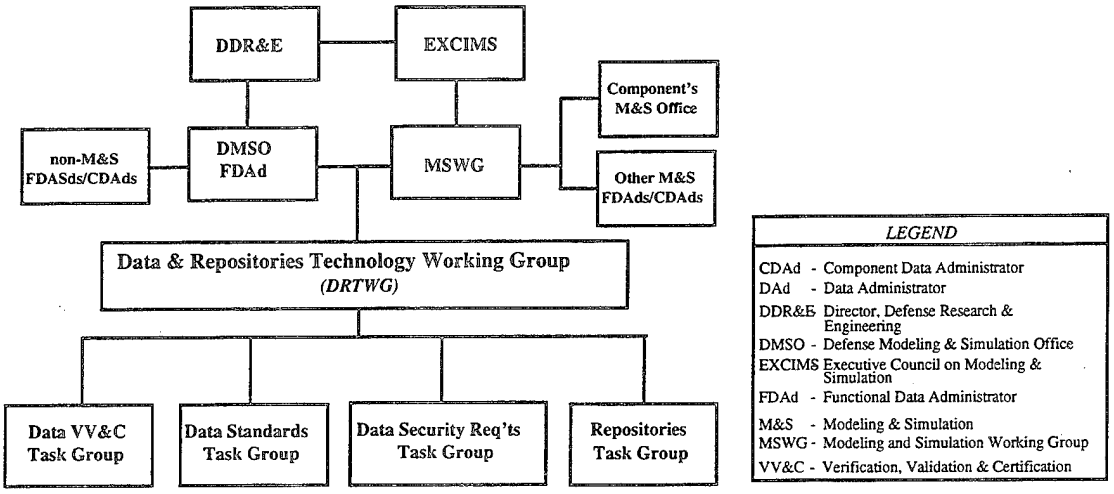


Figure 3. M&S DA Technical Infrastructure Support Groups

The M&S FDAd coordinates the M&S DA activities with Components in a number of ways through the organizational structure outlined in Figure 3. The M&S FDAd is assisted by members of the DRTWG drawn from the government (e.g., Services, Joint Staff, Agencies, NIST, NASA, OSD), federally funded research and development centers (FFRDCs), and contractors. The membership has grown to about 200 individuals since 1992. Through consensus building, these groups identify issues, define requirements, and develop recommendations for implementation. As a result, the M&S FDAd works closely with DRTWG members with functional and technical backgrounds to: (1) ensure that DA policy, procedures, and standards are being implemented effectively; and (2) develop standard data products for current and planned simulation applications.

3.2 Data Verification, Validation, and Certification (VV&C) Task Group

The data VV&C Task Group was formed based on the 1993 MORS Conference and has two Subgroups to address data quality.

3.2.1 VV&C Guidelines Subgroup. The VV&C Guidelines Subgroup has developed definitions for the VV&C of data. The definitions apply to (1) data developed specifically for M&S applications which requires that data VV&C be integrated with M& S VV&A (Reference 9), and (2) data produced generically for M&S and other functional areas such as C3I and logistics (example producers could be the Defense Mapping Agency's terrain data and the Defense Intelligence Agency's intelligence data). The data VV&C effort is also defining a data quality profile (metadata) (Reference 10) that describes the condition of the VV&C'ed database/data set, including the V&V procedures performed on it and audit trail information as to how the data was derived. Certification requires attachment of the data quality profile. This subgroup is in the process of developing a VV&C guidelines document based on ten studies of how data VV&C is performed in Service and DoD Agency organizations. Plans are to use the guidelines in one or two pilot studies during FY96-97 before refining them and developing data VV&C policy to implement them. Part of this effort is also concerned with defining metrics for measuring data quality.

Outstanding issue: There is high relevancy of the Data VV&C work toward addressing requirements for M&S VV&A. The condition and validity of the data used in M&S is a critical part of the VV&A process and has a profound effect on interpreting M&S results. This effort will need support in order for the guidelines to become DoD M&S policy and has potential as DoD-wide VV&C guidelines.

3.2.2 Authoritative Data Sources Subgroup (ADS). The ADS Subgroup has developed initial definitions for "data source", "authoritative data source", "data customer", and "data center". In addition, it has initially defined the roles and responsibilities of these entities. The subgroup has been actively developing a taxonomy or set of categories of M&S information areas in order to categorize authoritative data source entries, and to aid searching for an authoritative data source. The initial ADS Directory is currently available through the interim M&S Resource Repository (MSRR) on the World Wide Web (WWW).

Outstanding issue: This effort requires visibility at the DoD level since it is relevant to all aspects of information management across DoD. Identifying and obtaining consensus on these

authorities and giving producers responsibilities, including coordinating database development in their areas of authority, will be effective in reducing the number of stovepipe database development efforts. This can save dollars, avoid the need for future data adjudication between similar data collections, and provide quality data to M&S users.

3.3 Data Standards Task Group

The Data Standards Task Group has two subgroups addressing data standardization: the Distributed Interactive Simulation (DIS) (References 11, 12) Data Standards and Repositories Special Interest Group (DSR SIG), and the Complex Data Subgroup. Standard data is key to M&S for automated acquisition of input data and automated interoperability of models and credibility of simulation results. The Task Group will support M&S FDAd's development of data standards:

- To support common representations of data for use in models and simulations
- Maintain standard data elements and data models in distributed repositories that enable data sharing, reuse, and single point-of-entry
- Reduce need for data translation between applications, between databases, from message systems to applications, etc
- Enable advanced communications such as self-describing messages.

The M&S community is dependent upon the rest of the DoD data administration community for the majority of its data standards and does not standardize most of its data elements independently. This is because the M&S community is primarily a user of data defined and often produced, in other functional areas (Reference 14) and DoD Components. The M&S community uses international, national, Federal, and DoD standards. The M&S FDAd coordinates and guides the assembly, review, and submittal to the DoD Data Administrator (DAd) (Reference 14) of data model proposal packages from the M&S community. In addition, the M&S FDAd receives other functional data model proposal packages from the DoD DAd for distribution to selected reviewers in the M&S community.

The M&S community uses the evolving simulation architecture and DoD Enterprise Model (process and data models) as a framework to facilitate interoperability and reuse. The DoD Enterprise Model and the DoD Data Model provide a common perspective to support cross-functional integration and control data redundancy. M&S will integrate its data models with the DoD Data Model.

Planned activities in support of data standards include: building an automated process for developing M&S data standards and submitting them to DoD; facilitating the integration and extension of M&S data models with the DoD Data Model and the C2 Core Model; continuing to use, refine and extend the reverse engineering methodology (Reference 13) developed by DMSO to determine data requirements and standards from legacy databases and legacy M&S systems; and investigating object-oriented database techniques which lead to proposing appropriate extensions to DoD data modeling and DoD data standards to accommodate data objects.

Outstanding issues: Supply of data used by M&S must evolve from current stovepipe databases to intermediary data centers (e.g, TADS, OASIS) that provide aggregated data to models and

simulations, to provider data warehouses (e.g., Defense Mapping Agency, Defense Intelligence Agency) that will directly supply data to models and simulations. Essential to the data centers and warehouses is the use of data standards to achieve interoperability across M&S as well as for sharing and reusing data. Coordination across data warehouses can be accomplished procedurally and electronically (exchange of metadata and instance data in standard file formats) when interoperability across M&S data centers is a reality. There are hard issues to address, such as maintaining currency on derived databases versus database creation on demand. Since the unclassified DMSO MSRR node only supports unclassified directories, another issue, is the need for classified directories and the need for users to know about their existence.

3.3.1 DIS DSR SIG. The purpose of the recently formed DSR SIG is to make recommendations to the DIS community (DoD and non-DoD government, academia, and industry) on requirements for use of DoD Data Standards and M&S Resource Repository (MSRR), and accompanying rationale and guidance documents. More explicitly, the SIG will define DIS functional area requirements for data standards (i.e., data models, standard data elements, data dictionary) which will be compliant with DoD data standards (References 8, 14, 15, 16) and relevant commercial standards and define or identify common datasets (e.g., authoritative data sources) for use in DIS exercises.

Current DMSO activities in support of DIS include: developing a database containing the enumerated values for data items exchanged through DIS Protocol Data Units (PDU) (References 12, 17); initial development of a DIS Data Dictionary based on the data items described in the PDUs; and extension of the DIS Data Dictionary to include "more static" data items not present in the PDUs but found in the reference databases for the PDU entities. Other possible DIS DSR related activities include: IDEF process and data models of the DIS exercise life cycle; methods for an exercise planner to locate authoritative data; implementation of DIS data standards, symbology standards and complex data standards; implementation of a DIS node of the MSRR; DIS guidelines for configuration control procedures and toolsets to manage DIS resources; and DIS data security requirements.

3.3.2 Complex Data Subgroup. The M&S community deals with technical and scientific data that is often represented as complex data rather than atomic data. Complex data is difficult to model, standardize, and to share because it may be a composite data structure or be derived data (References 18, 19). The M&S FDAd coordinates the ongoing efforts to develop data models and standards for complex data, and promotes the incorporation of complex data standardization procedures into the DoD DA community.

Activities in support of complex data include: pilot studies of highly derived complex data (such as weapon performance probabilities hit/kill) in order to develop improved data modeling tools and metadata to describe data standards; and proposed improvements and extensions to IDEF1X to handle modeling of complex data and to DoD data standards to handle extensions to metadata definitions.

Outstanding issue: Much of the data that is shared and reused by the M&S community is scientific and technical data that is not being addressed by DoDD 8320.1-M-1 (Reference 15). Therefore, standardization of complex data must receive high priority when addressing

requirements. For example, the M&S community needs to describe data standards in the data dictionary for concepts such as probability of kill or hit (P_k and P_h) in such a way that M&S developers and users can quickly understand how these were derived and what standard is relevant to their problem.

3.4 Data Security Requirements Task Group

Data security is becoming critical to M&S because of the need to run large exercises (e.g., DIS, STOW) that include both classified and unclassified sites, and the planned implementation of distributed unclassified and classified MSRR systems. Even for the unclassified interim MSRR, there are issues of access control and user authentication, as well as releasability and sharing of information resources. Conflicting security policies and procedures, and changing requirements and technologies, are forcing DoD to take a critical look at data security.

The Task Group is currently carrying out activities to define the M&S data security requirements for technology and policy. A very critical near term effort is to support the unclassified interim MSRR needs to control user access and to support sharing and releasability of information resources among the M&S community.

Outstanding issues: Addressing the data aggregation security classification issue is of high importance because of the frequent need to aggregate large amounts of data for M&S. An aggregated M&S database may require a higher security classification than the individual data sets. Once the aggregated database is created, it is difficult to release subsets of data at the lower classification level.

The need for multilevel secure (MLS) data management is of high relevance to M&S since there will be a need to gather data from databases at different classification levels to run large exercises. Currently, both unclassified and classified MSRR systems are being planned pending maturation of MLS technology.

Releasability of data is an issue for unclassified as well as classified M&S data. While there is DoD security policy protecting data, there is no DoD policy supporting the sharing of data except for the Freedom of Information Act. As a result, it is currently easier and safer for DoD release authorities to prevent or delay release than to support it. While it costs nothing to not share, there may be costs associated with sharing such as preparing datasets for export, explaining the data, etc. The M&S Security Task Group will address releasability policy. Associated with this are considerations of how to provide fee for service.

3.5 Repositories Task Group

M&S employs and promotes the use of common procedures and tools to support the M&S FDAd, study directors, data administrators, and the functional and technology support community. These procedures and tools provide shared access to standard data products (e.g., process and data models, complex data representations, and data element definitions). They also support reusability and interoperability of associated M&S resources (e.g., metadata, data, algorithms, models, simulations, and tools) among developers and users throughout the M&S community.

The Repository Task Group worked with the DoD DAd to prioritize changes to be implemented in Defense Data Dictionary System (DDDS) releases, and with the DoDR Steering Committee to identify M&S community's requirements for the DoDR. M&S requirements include a repository metamodel, directories, support for complex data, data VV&C and data security, authoritative data sources, configuration management, data collection and distribution (e.g., data centers), and support for the FDAd and DoD data standardization process.

3.5.1 M&S Resource Repository (MSRR). The M&S FDAd is implementing a system of distributed MSRRs to efficiently and effectively provide the M&S community with data, metadata, algorithms, models, simulations, and tools. The MSRR concept of operations (CONOPS) addresses operation and management; policies and procedures; information resource management and protection; technology insertion and standards; and security. The MSRRs will also maintain and provide quality assurance information (e.g., model assumptions, data source, data classification, range of validity of algorithms, VV&A/C status, and configuration history) to improve the usability and credibility of modeling and simulation. Additional tools provide for developer and end-user access, browsing, and retrieval of M&S resources from the MSRRs.

The M&S FDAd will establish distributed interim MSRR (iMSRR) nodes with access via Internet WWW server nodes for collection, maintenance, and retrieval of data products and M&S resources. This initial capability is intended to establish and enhance communication coordination and information sharing between DoD M&S activities and allow a more disciplined evolution to the objective MSRR. The initial MSRR node is at DMSO and is available on the WWW (http://www.dmso.mil).

An Implementing Committee will collaborate and concur on an MSRR architectural framework of standards and conventions (compatible with the Technical Architecture Framework for Information Management (TAFIM); policies and procedures; and the organization and responsibilities of MSRR groups (i.e. Registrar, Users Working Group, Node Administrators Working Group, etc.) The first meeting will be held in August 1995. Much of the MSRR architecture will be based on similar operational systems such as the Intelligence Community's Intelink and NASA's Earth Observing System Data and Information System (EOSDIS).

Current activities include: participation as a DoDR beta test site for the to-be-selected new DoD repository software; providing M&S Community directories to databases, models and simulations and authoritative data sources on the iMSRR; establishment of DMSO sponsored iMSRR nodes for each Service and the Joint Simulation System program by the end of FY96; development of configuration management procedures and tools for supporting management of MSRR resources; and a full plan for evolving from the iMSRR to a comprehensive, fully operational DoD-wide system of distributed MSRRs by FY99.

Outstanding issue: It is important to meet incremental requirements in developing the MSRR to reach the final goal of seamless access to M&S information resources. The iMSRR will a base for all future M&S repositories. Its goal is migration of existing resource repositories, and seamless or near seamless access to many other information resource repositories throughout unclassified and classified networks. In development, the technical risk is low through the prototype phase and becomes riskier during the operational phase because of uncertainty about

the availability of repository standards and COTS products to implement the standards. If the future DoD Repository is available and is selected for the MSRR, the risk may be lower. Once the MSRR is operational there may be high risk in the final phase if multilevel security is implemented.

4. SUMMARY

This paper describes the Defense M&S DA Program's mission, scope, implementation approach, goals and objectives, and action plans to achieve the DoD M&S Master Plan objectives.

The major accomplishments of the M&S DA Program from FY93 to date, listed below, form the baseline for future phases of the M&S Master Plan.

- The M&S FDAd established the M&S Data Administration Program and developed close coordination with the Component M&S offices to provide data administration support to the M&S community.
- The DRTWG identified key data issues being addressed in the data portions of the M&S Master Plan and supporting Investment Plan to guide DMSO's short-term and long-term DA initiatives.
- The M&S FDAd provided M&S DA services for a pilot study in modeling complex data, submitting and reviewing candidate standard data elements to DoD for approval, training users, facilitating development of shared databases and reusable data. Also the M&S FDAd developed and validated the reverse engineering methodology for several simulation systems, published the handbook for migrating legacy M&S databases (Reference 13), and supported the development of data quality engineering tools.
- The M&S FDAd facilitated M&S information sharing by hosting DRTWG conferences and working group meetings, and presenting papers at MORS conferences, DIS workshops, CENTCOM's M&S Data Base Conferences, an Intelligence community M&S Symposium, and M&S Industry Days.
- The M&S FDAd developed the initial node of the interim MSRR in support of the future DoD Repository System.

Major emphases of the M&S 1996-2000 DA are listed below:

- In accordance with the DoD Corporate Information Management/Enterprise Integration (Reference 20) and the M&S Master Plan, develop a data technical framework and extend the M&S infrastructure to support developers' and end users' needs.
- Develop data standards to support common representations in models and simulations.
- Leverage new technologies through R&D activities to include object-oriented database management and distributed databases.
- Establish methodologies, standards, and procedures for the VV&C of data as part of the M&S VV&A process to support credible M&S results.
- Define specific M&S data security requirements for access across repositories.

- Provide classified and unclassified distributed MSRRs to facilitate developer and end-user access to M&S information resources for reuse and sharing.

5. ACKNOWLEDGMENTS

The authors wish to express their appreciation to the DRTWG team members, especially to the co-chairs of the Task Groups and Subgroups who have made such remarkable progress over the past eighteen months. The authors also wish to acknowledge the comments and encouragement received from CAPT J.W. Hollenbach, COL J. Wiedewitsch, USA; CAPT(Sel) M. Lilienthal, MSC, USN, Ph.D.; D. Cantrell, H. Haeker; and administrative support from Robert Senko, Linda Lange, and Cheryl Homatidis. Lastly, the authors wish to express their appreciation to the support received from DMSO, DISA/Center for Standards and RAND.

AUTHOR BIOGRAPHY

Dr. Chien Huo is a Computer Scientist with the U.S. Army and is also an Associate Director of the Defense Modeling and Simulation Office (DMSO). He is currently the Point of Contact for the M&S Functional Data Administrator and is responsible for DoD M&S data administration. He co-chairs the M&S Data & Repositories Technology Working Group (DRTWG) and its associated task groups on Data Standards, Data Security Requirements, and Repositories. He is a member of AIAA, IEEE, Sigma Xi. He served on the AIAA Flight Simulation Technical Committee. Dr. Huo received his B.S. in Mechanical Engineering from the National Taiwan University in 1968, an M.S and Ph.D. in Applied Mathematics from Brown University in 1973 and 1974. Questions on this paper may be addressed to him at: DMSO, 1901 N. Beauregard St., Suite 504, Alexandria, Virginia 22311; Telephone - (703) 998-0660; Fax - (703) 998-0667; or Internet - msfdad@dmso.dtic.dla.mil.

Ms. Iris M. Kameny is an Associate Director of RAND's Acquisition and Technology Policy Center, responsible for information science projects including the areas of intelligent databases, simulation, concurrent processing, and knowledge based systems. She co-chairs the M&S Data & Repositories Technology Working Group (DRTWG) and its associated Task Groups on Data Standards, Data Security Requirements, and Data VV&C. She is a member of the Army Science Board, the Military Operations Research Society (MORS) Senior Advisory Group on Simulation Data, IEEE, ACM, AAAI, and AIAA. Ms. Kameny received her B.A. in Psychology from the University of California, Los Angeles in 1954. Questions on this paper may be addressed to her at: RAND Corporation, 1700 Main Street, Santa Monica, California 90407-2138, Telephone - (310) 393-0411, X7174, Fax - (310) 393-4818; or Internet - kameny@rand.org.

REFERENCES

[1]DMSO, "Defense Modeling and Simulation Initiative," May 1992.

[2]Deputy Secretary of Defense Memorandum; "Accelerated Implementation of Migration Systems, Data Standards and Process Improvements," October 13, 1993.

[3]Assistant Secretary of Defense Command, Control, Communications, and Intelligence (ASD(C3I)) Memorandum; "Selection of Migration Systems," November 12, 1993.

[4]Principal Under Secretary of Defense (Acquisition and Technology) Memorandum; "Implementation of Deputy Secretary of Defense Memorandum," January 11, 1994.

[5]ASD (C^3I) Memorandum; "Accelerating DoD Data Standardization and The Rapid Data Standardization Guidance," May 23, 1994.

[6]DoD 5000.59-Paa, draft; "Defense Modeling and Simulation Master Plan," January 1995.

[7]DoD 5000.59; "DoD Modeling and Simulation (M&S) Management," January 4, 1994.

[8]DoD 8320.1-M; "DoD Data Administration Procedures," March 1994.

[9]DoD Instruction 5000.xx; draft, "DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A)," July 3, 1995.

[10]J. Rothenberg, RAND Report No. DRR-1025-DMSO; draft, "A Discussion of Data Quality for Verification, Validation, and Certification (VV&C) of Data to be Used in Modeling," April 1995).

[11] C. Huo, "An Activity Model for Standards Process for the Distributed Interactive Simulation," Winter Simulation Conference Proceedings, December 15-19, 1993, Los Angeles CA., pp 1013-1020.

[12] IST-SP-94-01, "The DIS Vision, A Map to the Future of Distributred Simulation," May, 1994.

[13]DoD 5000.bb-M; "DoD Joint Data Base Elements For Modeling and Simulation (JDBE) Methodology Handbook", draft, February 1995.

[14]DoD Directive 8320.1; "Department of Defense Data Administration," September 26, 1991.

[15]DoD 8320.1-M-1; "Data Element Standardization Procedures," January 1993.

[16]DoD 8320.1-M-x; "DoD Data Model Entity Procedures," April 1993.

[17]IEEE Standard #1278; "Protocol Data Unit Standards for Distributed Interactive Simulation Applications," May 1993.

[18]I. Kameny and C. Huo, DISA/JIEO/CFS Report #94-01; "The Study Report for The Complex Data Modeling Workshop for TADS Weapon Performance Data," (TRAC, Ft. Leavenworth, KS, August 16-20, 1993), September 14, 1993.

[19]DMSO I/DBTWG Complex Data Task Force Report; "Complex Data Categorization," April 6-7, 1994.

[20]ASD (C^3I) DoD Strategic Plan and Implementing Strategies, draft, "Corporate Information Management/Enterprise Integration for the 21st Century," November 1994.

COMNAVSECGRU

Data Administration and Data Standardization

Data Base Colloquium

28 August 95

Duane L. Waggoner
CNSG Data Base Administrator
STU-III:(202)764-0430 DSN:764-0430
Fax (202) 764-2914

Outline

- **Background**
  - What is Data Standardization, and Why Do I care?
  - Development of NSG Master Data Element Dictionary (MDED)
  - NWTDB Standards and Structures Manual
- **Current Activities**
  - NSG MDED
  - Common Cryptologic Database (CCDB)
  - NSA Coordination
- **Future Activity**

UNCLASSIFIED

What is "Data Standardization", and why do I care?

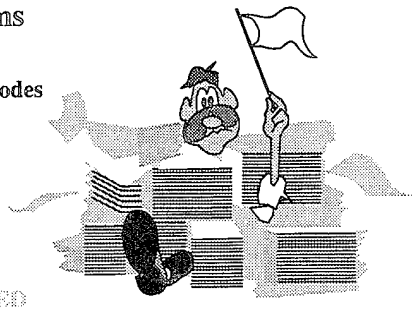
- Data standardization is the process of reviewing and documenting the names, meanings, and characteristics of data elements so that all users of the data have a common shared understanding of it.

- Promotes Interoperability
- Provides developers the basic building blocks required to develop software systems
- Compliance Mandated by OSD

UNCLASSIFIED

The Problem With Existing Databases

- Operational Problems
  - Redundant data
  - Redundant processing
  - Complexity and interdependence
  - Difficult to change (Legacy Systems)
- Information Related Problems
  - Inconsistent data
  - Inconsistent representations and codes
  - Lack of understanding
  - Data quality



UNCLASSIFIED

Comparison of existing Data Elements

Name	*Source Document*	*Domain Value*
Antenna Type	USMTF	1-17 A
	Manual of Standard Data Elements and Related Features	2 A
	NWTDB	3 CHAR
Ship Type	MIIDS/IDB	1-6 A
	IDEAS	3 N
	JOPES	4 N
	USMTF	1-8 A

UNCLASSIFIED

The Solution

- Gather and register data requirements
- Model your data to a common standard (DoD Enterprise Model/C2 Core Data model)
- Build and maintain data dictionaries
  - Two types of dictionaries
    - » Active
      - dictionary and DBMS are integrated. DBMS uses definitions at run time
    - » Passive
      - dictionary and DBMS data definitions are separate

UNCLASSIFIED

Background

- NSG developed Master Data Element Dictionary to aid systems developers in development of cryptologic systems to achieve interoperability
- Joined NWTDB process as the FDBM for Cryptology

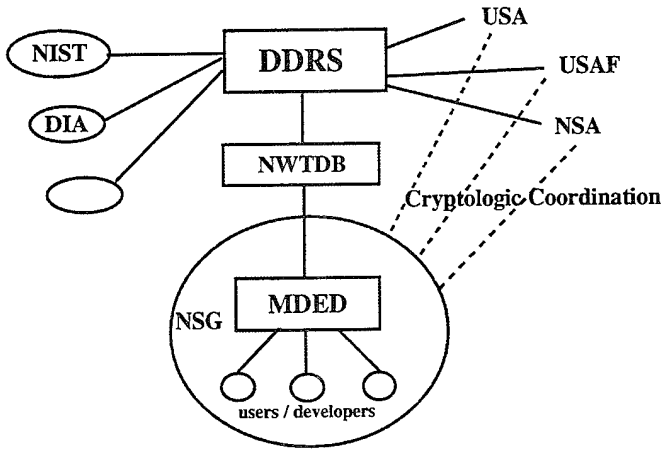
UNCLASSIFIED

Relationships

- Navy - CNO, ONI, NISMC
- DIA
  - MIIDS/IDB
- NSA
  - Data Administration
  - Center for Standards
- USAF
  - CONSTANT WEB
- USA
  - TEARS

UNCLASSIFIED

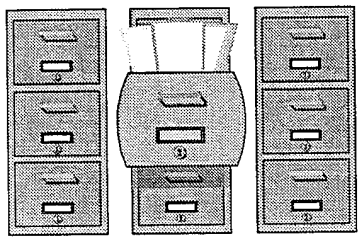
Master Data Element Dictionary



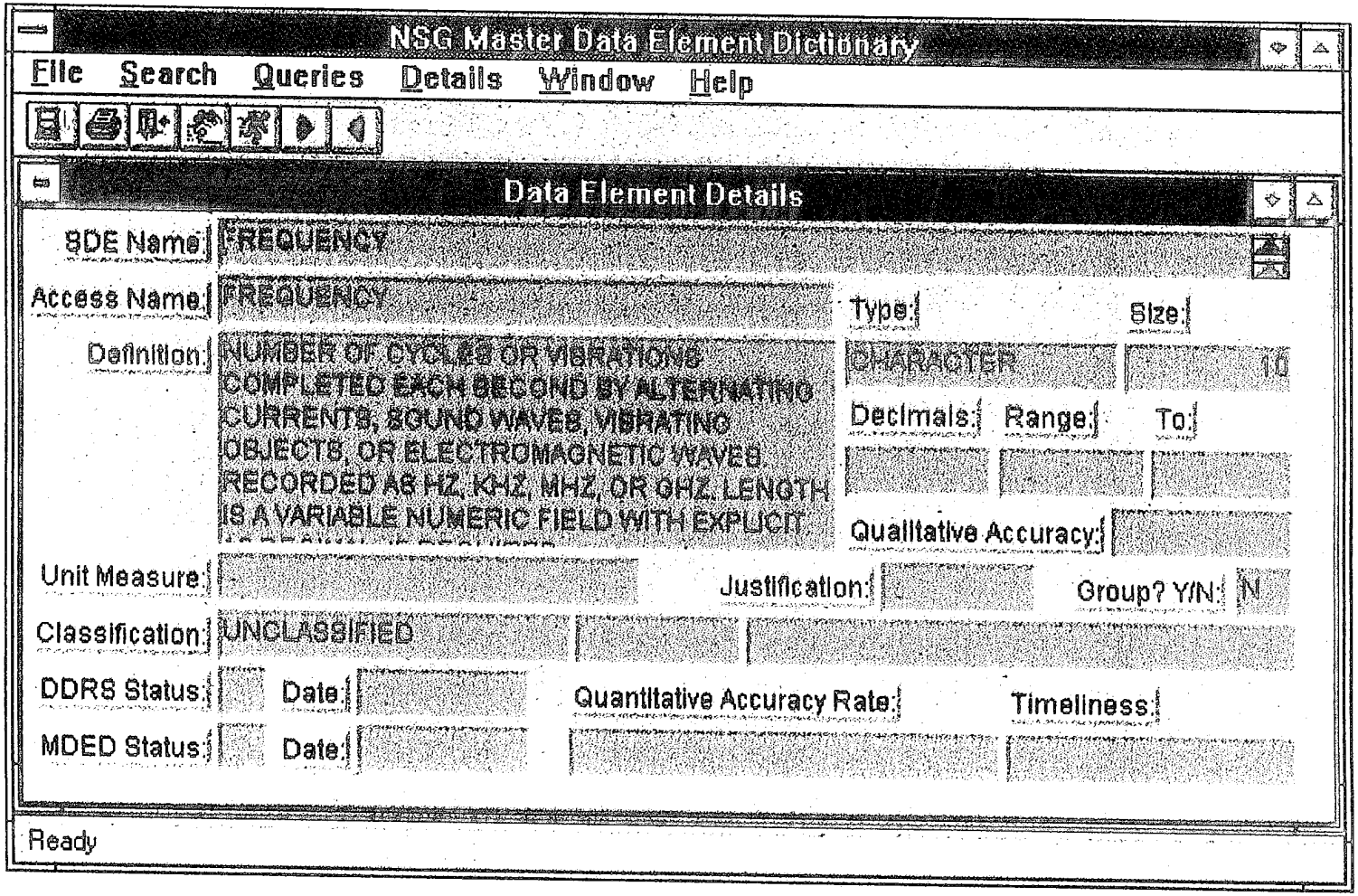
UNCLASSIFIED

NSG MDED

- DISA Compliant (DODDIR 8320.1 series)
- Built in Oracle 7 running on Vax 4400 under VMS
  - Oracle Forms v3.0 used for interface
- Repository of NSG standards data elements
- Centrally managed at HQ Naval Security Group Command
- Approximately 2500 terms
- Passive Data Dictionary
- Adopted by NSA
  - being converted to run on SPARCs
  - developing GUI



UNCLASSIFIED


NSG Master Data Element Dictionary
File Search Queries Details Window Help
Data Element Details
SDE Name: FREQUENCY
Access Name: FREQUENCY
Type: CHARACTER
Size: 10
Definition: NUMBER OF CYCLES OR VIBRATIONS COMPLETED EACH SECOND BY ALTERNATING CURRENTS, SOUND WAVES, VIBRATING OBJECTS, OR ELECTROMAGNETIC WAVES. RECORDED AS HZ, KHZ, MHZ, OR GHZ. LENGTH IS A VARIABLE NUMERIC FIELD WITH EXPLICIT
Decimals:
Range:
To:
Qualitative Accuracy:
Unit Measure:
Justification:
Group? Y/N: N
Classification: UNCLASSIFIED
DDRS Status:
Date:
Quantitative Accuracy Rate:
Timeliness:
MDED Status:
Date:
Ready

"Developer's" MDED

- PC Windows-based application
  - 386 with minimum 4 MB RAM (8 MB preferred)
  - MS-Windows
- Application developed using Powerbuilder
- uses DBASE file structure
- contains all 2500 terms from Oracle version
  - only attributes pertinent to developers are displayed
- only used as a browser
- distributed via CD-ROM or 3.5" disks
- submission of new terms via hardcopy or softcopy

UNCLASSIFIED

MDED Distribution

- SPAWAR document references the MDED for all Naval Cryptologic Systems development
- Copies sent to contractors supporting SPAWAR cryptologic development efforts
- Copies of MDED requested and sent to NAVSEA, NSA, NRAD, other government agencies

UNCLASSIFIED

Future

- Continue to work on NWTDB Standards and Structures manual
- Continue development of NSG Data Model
- Coordinate submission of Cryptologic Data Elements through NSA to DDRS
- Assist NSA and FDAd - Intelligence with data repository

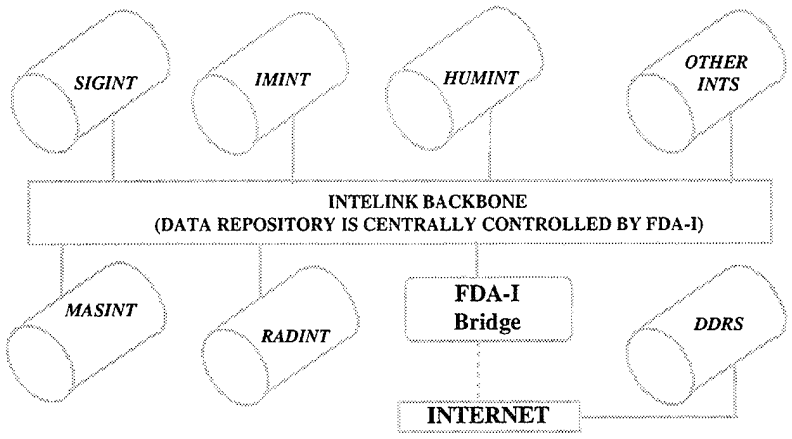
UNCLASSIFIED

Secure Data Repository

- Distributed database structure
  - Common data dictionary structure used by all data stewards
- Relational database (ORACLE, SYBASE, etc.)
- Accessible via INTELINK
- Common GUI being developed
- Data stewards maintain only the data elements attributed to them
- Access to DDRS will be provided through FDAd-I
  - via air gap until trusted Multi-Level Secure system is delivered
- Ultimate goal is to provide an active dictionary

UNCLASSIFIED

Proposed Secure Data Repository



UNCLASSIFIED

So why is Data Standardization important?

- **Realize cost savings**
- **Migration of legacy systems**
- **Improve Interoperability**
- **Improve C4I for the warrior**
- **Compliance with DoD Directive 8320 series**

UNCLASSIFIED

Author's Biography

Duane L. Waggoner is the Database Administrator for the Commander Naval Security Group (CNSG). He sets the policy for database development efforts within CNSG and its field sites to ensure interoperability and standards. Mr. Waggoner is the Cryptologic Functional Database Manager (FDBM) for the Naval Warfare Tactical Database (NWTDB) and attends other working groups such as Naval Information Systems Management Center Data Administration Network (NISMC DANET) and Joint Maritime Command Information System Data Engineering Working (JMCIS DEWG). He has over twenty years experience in the database development and computer field. Mr. Waggoner has been on the Database Colloquium committee for the last seven years representing CNSG. He has a B. S. degree in Information Systems Management from University Maryland University College.

"Achieving Joint Interoperability through Information Standards"

LTC John D. Burke

Office of the Director for Information Systems,
Command, Control and Computer Systems (DISC4)
Headquarters, Department of the Army

1. INTRODUCTION

The US Department of Defense has 40,000 or more information, weapon, and Command and Control (C2) systems under development, migration, or designated as legacy systems. Perhaps at least as many software systems exist as "command uniques" developed by individual major commands and activities to satisfy a unique or time critical need. These information systems range in importance and sensitivity from satellite intelligence imagery to the location and waiting time for installation housing. Unfortunately, even though each was originally designed to meet a specific functional need and an implied or specific performance criteria, the traditional outcome has been to evolve into "stovepipe" systems that lack even basic interoperability between systems of like nature and mission.

Taken beyond the context of narrow scope systems, there exists a compelling need to satisfy the competing requirements of communications frequency bandwidth, heterogeneous databases, different switching systems, multi-level security, and multiple programming languages. Each area represents a significant interest in industry and DoD to improve the efficiency and utility for all types of data. Data becomes the common denominator for the transfer of information between users regardless of the platform, operating system, and communications media. It is persistent across applications and represents a common building block throughout requirements analysis, design, development, testing, and fielding.

2. JOINT INTEROPERABILITY IN CONTEXT

Joint Interoperability Requirements and Doctrine

The authoritative source of joint interoperability, Command and Control Computer Systems for the Warrior (C4IFTW) states, "At the center of the C4IFTW concept is the establishment of a global C4I capability that allows the warfighter to plug in anytime, anyplace, in the performance of any mission." Clearly, the underlying theme in discussions regarding joint interoperability rotates around the need to leverage and apply technology in support of the warfighter and the military missions. The need for successful joint interoperability may be demonstrated by the volume and importance to the theater commander or Joint Task Force (JTF) commander's reliance on information.

General Powell, in his assessment of the Persian Gulf conflict, reported to Congress, "At the height of the Persian Gulf conflict, the automated message information network passed nearly

2 million packets of information per day through gateways in the Southwest Asia theater of operations. The technology developed to support these networks proved to be a vital margin that saved lives and helped achieve victory." Furthermore, the information being passed included much more than simply data. Voice and network management played a key role in ensuring reliable information transfer over the available communications network was feasible. The DoD Final Report to Congress: Conduct of the Persian Gulf War, April 1992, assesses that, "At the height of the operation, this hybrid system supported more than 700,000 telephone calls and 152,000 messages a day. Additionally, more than 35,000 frequencies were managed and monitored daily."

Required C4 Capabilities

Our joint warfighting publication for Command, Control, Communications, and Computer (C4) Systems Support to Joint Operations, lists these mandatory C4 capabilities:

1. Support activities across the range of military operations
2. Support a smooth, orderly transition from peace to war.
3. Provide for the collection, processing, transmission, and dissemination of data and products.
4. Protect systems/networks through C4 defensive measures.

This paper will focus primarily on the third key doctrinal mandate for joint interoperability, the dissemination of data and products. The purpose of the required C4 mandates is to support the objectives of the C4 systems to meet the needs of the warfighter. These needs may be summarized as follows:

1. Produce a Unity of Effort through common understanding by multiple commanders.
2. Exploit Total Force Capabilities by extending human senses and processes.
3. Properly Position Critical Information in order to respond quickly to a request for information and maintain the information where it is needed.
4. Information Fusion in creating a common picture of the battlefield that is accurate and meets the needs of the warfighters.

The emphasis on joint interoperability must be considered beyond data. The fourth tenet of C4IFTW, Horizontal and Vertical C2, states that the communications requirements, "... may be data, voice, video, or integrated mode. The goal also includes doctrine, standards, terminology, and data availability and processing to ensure common interpretation and understanding." The current efforts in DoD data standardization support the need for common interpretation and understanding by creating a set of information standards that capture the entire function and relationships between data supporting different areas of interest.

3 Levels of Operations

Joint Interoperability is achieved through the fusion of three levels of operations: Doctrinal Interoperability, System Interoperability, and Operational Interoperability. These levels of operations represent the perspectives of the joint planner, the C4I staff, engineers, and software developers, and the task force units which must employ land, sea, and air forces to achieve the mission.

Doctrinal interoperability is achieved through the collaboration of the individual service requirements, capabilities, and warfighting philosophy. JCS Pub 6-0 states a number of principles for achieving joint interoperability. These include, Establish Liaison Early; Effective Use of Limited C4 Resources such as space based assets; JCSE, and frequency spectrum; Standardization of Principles and Procedures; Agreement in Advance of War for the principles, procedures, and overall communications requirements to include standard message text formats; *STANDARD DATA* bases and formats; and frequency management.

System interoperability is the mechanical, physical, electrical, and communications interfaces that allow one weapon, C2, or information system to pass information from user to user without human intervention. In terms of communications systems, joint doctrine states that C4 systems include these key components: 1) Terminal Devices such as telephones, fax machines, and computers; 2) Transmission Media to include radio, space based systems, wire and fiber; 3) Switches, either circuit or message (packet) that route traffic through the network; and 4) Control in terms of network or nodal. Network control provides management of networks, while nodal control manages the local C4 networks and equipment. The key point is that system interoperability cannot be achieved by design or software alone; a physical medium complete with the design, production, and operational constraints, must be included in the overall assessment of system interoperability.

Operational interoperability is the condition of fighting in a joint or combined situation where each force package is able to link functionally, and through its C2 systems into a single operating unit. This type of interoperability is documented in the relative joint and service doctrines such as the Army Field Manual 100-5, "Operations", and Training and Doctrine Pamphlet, 525-5, "Force XXI Operations."

Army's Force XXI

The premise of the Army's Force XXI is the need to use technology and information to achieve victory over the enemy by being able to know the enemy's intent and dictate terms for which he is unable to react in time. Information is a pervasive and powerful part of Force XXI since it is expected that, "Information technology is expected to make a thousandfold advance over the next 20 years." The information war is based on two key parts. First, information technology will greatly increase the volume, accuracy, and speed of battlefield information available to commanders. Such technology will allow organizations to operate at levels most adversaries cannot match while simultaneously protecting that capability. Second,

advances in information management and distribution will facilitate the horizontal integration of battlefield functions and aid commanders in tailoring forces and arranging them on land.

Force XXI expects a complete change in the concept of fighting the battle where it will change from one with a rigid hierarchy to that where, "Individual soldiers will be empowered for independent action because of enhanced situational awareness, digital control, and a common view of what needs to be done." Each individual soldier thus becomes a separate battlefield component capable of observing, orienting, deciding on the appropriate action, and acting on that information. Yet, to prevent the traditional stovepiping of individual systems within a Service, the US Army doctrine expressly ensures that, "Joint and Multinational operations will be facilitated by improvements described in the battle dynamics, early twenty-first century American land operations will be fully integrated, completely joint, and most often multinational."

DoD Guidance on Interoperability

DoD guidance and direction provides some operating guidelines for the assurance of joint interoperability. These include two key DoD Directives. DoD 4630.5, Compatibility, Interoperability, and Integration of Command and Control, Communications, and Intelligence (C3I) Systems and DoD 4630.8, Procedures for Compatibility, Interoperability, and Integration of Command, Control, Communications, and Intelligence (C3I) Systems.

The identification of an C2 or information system by Service is now passé. The DoD Directive 4630.5 states, "It is DoD Policy: That for purposes of compatibility, interoperability, and integration, all C3I systems developed for use by US forces are considered to be for joint use." This policy has become the source for several other related DoD and DISA policies on joint interoperability certification and testing.

The DoD Directive 4630.8 identifies the responsibilities of the Chairman of the Joint Chiefs of Staff who shall: 1) "... confirm that interoperability requirements for interfaces, software integration with other C3I or supporting functional information systems, ... are clearly identified in requirements submissions for new or modified C3I capabilities;" and, 2) "In accordance with (Defense Acquisition Policy), validate interface standards and recommend to the ASD(C3I) for approval as appropriate." Furthermore, "All C3I systems and equipment shall conform to technical and procedural standards for compatibility and interoperability, developed or recommended by the DISA under guidance provided by the CJCS."

The achievement of joint interoperability cannot be done solely through policy and procedure. The inherent capabilities within weapon systems and their design must represent the warfighter's intention and concept of operations. The systems to execute joint interoperability are, by their nature and the complexity of using multi-service systems, also complex, highly integrated systems. In essence, systems of systems. It is this concept of banding multiple systems of different designs, capabilities, and stages in life cycle, that mandates for successful joint system interoperability require that a structured systems engineering approach be used to improve the reliability of information exchange and information flow between systems.

3. SYSTEM ENGINEERING FOR C3I SYSTEMS

"Be quick, be quiet, be on time. If you can't do it with brainpower, you can't do it with manpower-overtime" quoted by "Kelly" Johnson, of the famous Lockheed Skunk Works, who was the Lockheed Blackbird (SR71) program manager. Prudent engineering management requires a well thought out design and validation of the design prior to development.

The next logical step in the achievement of joint interoperability is translation of the requirements and mandates described above into a design and structure capable of meeting these needs within the constraints of cost, schedule, performance, and supportability. The scale and breadth of joint interoperability indicates that even at the modular level, this is not a trivial task. To translate specified requirements we must engineer at the system level using structured analysis and design as a basis.

System Engineering Defined

One of the leading authors in the area of system engineering, Dr. Sage of George Mason University, has defined system engineering consistent with MIL STD 499A as, "the application of scientific and engineering efforts to (a) transform an operational need into a description of system performance parameters and system configuration; (b) integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces to achieve optimization; and, (c) integrate reliability, maintainability, human engineering ... within cost, schedule, supportability, and technical performance objectives."

However, he points out that there are problems in implementing this simple definition when dealing with interoperable and complex systems. There are distinct problems associated with the production of functional, reliable, and trustworthy systems of large scale and scope. He identifies three distinct areas of concern: 1) It is very difficult to identify the user requirements for a large system; 2) Individual new subsystems often cannot be integrated with legacy or heritage systems; 3) Large systems often do not perform according to specifications; and 4) System specifications often do not adequately capture user needs and requirements.

Dr. Dimitris Chorafas, in his book, "Systems Architecture and Systems Design", identifies these conditions as particularly problematic in large scale information systems engineering:

1) heterogeneous equipment
2) incompatible operating systems
3) installation in multiple locations
4) requirements for a timely response to user needs
5) transfers of large file volumes (text, data, graphs, voice, image)
6) increasing stress on reliability and availability

Requirements Analysis

The underlying need to understand and represent the user's requirements cannot be underestimated. A clear definition becomes the foundation for virtually all of the life cycle program documentation and the resultant test and evaluation criteria. Poorly defined requirements permeate the program's documentation requiring constant correction and redefinition throughout the engineering and production phases. It is equally unrealistic to think that all possible requirements can and will be understood and articulated at the beginning of the project. An encyclopedia of documentation is both unwieldy and unrealistic. Dr. Rechtin, in his book, "Systems Architecting," tell us that, "Amid a wash of paper, a small number of documents become critical pivots around which every project's management revolves."

Rechtin further recognizes the need to join the user and the designer together with at least a working understanding of the requirements. In fact, "... the de facto initial step in the development of complex systems is for the client and architect to take whatever requirements do exist and construct, through discussion, a rough model of a system that might satisfy most of them." The building of a model, even if only an abstract representation, needs to be as complete as is feasible since, "In the model building process, some of the original requirements are, or can be, lost or deferred. If unrecognized in the beginning, the losses can later be the cause of disappointment, frustration, and recovery costs." However, as we will explore later, there are measures to reduce the risk of incomplete requirements analysis through the use of DoD tools.

Utility of Modeling and Simulation

The Vision, a document that provides the Army Chief of Staff's perspective on the Army Enterprise Strategy, lists 10 principles essential to reaching the Force XXI capability. Modeling and Simulation is one of these 10 principles. The Army's Enterprise Implementation Plan enforces the use of models and simulation by requiring that, "The operational architecture, as the operational requirements components of the enterprise architecture, must be derived from the documents and models produced in the requirements process." Additionally, in the procedures to execute this task, "The process and data of the simulation are captured in IDEF and the simulation, using whatever methodology, technique, platform and language ..."

4. REQUIREMENTS (PROCESS) AND DATA MODELING

We have seen up to this point that a framework exists for conducting joint interoperability with broad operational goals and limited procedural constraints. Furthermore, the recognized methodology for beginning the system engineering of these requirements lies in a formal method of modeling and simulation. Within the Army, the two techniques required for activity (requirements) and data modeling are IDEF0 and IDEF1X, respectively.

The Department of the Army C4I Technical Architecture

The C4I Technical Architecture represents the strategy for implementing a multitude of standards and procedures to achieve interoperability within the Service as well as between

Services. "It is intended to serve not only the Army Enterprise Strategy implementation, but to be expanded and proposed to support the Joint Warfighter community." Additionally, the C4I Technical Architecture is mandated for use by the Army Acquisition Executive who has decision authority and oversees all army acquisition, research and development. The tools and techniques for interoperability are therefore included at the inception of any program and throughout its life cycle.

The C4I Technical Architecture applies to all tactical, strategic, and sustaining base information systems as well as all soldier, weapon, and information system programs. Materiel and combat developers will ensure this architecture is the basis for design and implementation and for determining performance and sustainment criteria. Combat developers will use the C4I Technical Architecture in developing requirements and functional descriptions. The C4I Technical Architecture consists of four parts: 1) Information Processing Standards; 2) Data Transport Standards; 3) Information Standards; and 4) Human Computer Interfaces. Each of these areas are briefly summarized below:

Information Processing Standards: The government and commercial standards that comprise the Common Operating Environment (COE) and the mandated use of common products that form the COE.

Data Transport Standards: Describes the data transport standards and profiles that are essential for data transport interoperability and seamless communications. Mandates the use of the open-systems standards used for the Internet and Defense Information Systems Network (DISN).

Information Standards: Information modeling using IDEF0 process modeling and IDEF1X data modeling. Describes the use of the Defense Data Repository System and the Command and Control (C2) Core Data Model.

Human Computer Interfaces (HCI): Specifies the HCI elements and the development guidance, mandates, and standards.

IDEF0 Process Modeling

Solvberg, in his book, "Information Systems Engineering," describes the beginnings of system design as the identification of requirements and a model of those requirements. The use of the model is to provide a common ground for all participants in the project. "The objective of the work in the system modeling and evaluation phase is to develop a logical model of functions and data that is sufficiently detailed to give users, management, and developers a realistic understanding of the properties of the system and its implications."

Federal Information Processing Standards Publication (FIPS) 183, "Integration Definition for Function Modeling (IDEF0)", states that, "Functional models produced through the IDEF0 technique provides a structured representation of the functions, activities or processes within the modeled system or subject area."

IDEF0 has the following characteristics:

1. It is comprehensive and expressive, capable of graphically representing a wide variety of ... enterprise operations to any level of detail.
2. It is a coherent and simple language, providing for rigorous and precise expression, and promoting consistency of usage and presentation.
3. Enhances communication between systems analysts, developers, and users through ease of learning and its emphasis on hierarchical exposition of detail.
4. Well tested and proven.
5. Can be generated by a variety of computer graphics tools.

The two primary modeling components are functions (represented by boxes) and the data and objects that inter-relate those functions (represented by arrows).Each activity, i.e., "Conduct Joint Operations", is bounded by four arrows: Input = those items which are transformed or consumed by the function (activity); Output = data or objects produced by the function; Controls = conditions (constraints) required for the function to produce the correct outputs; and Mechanisms = means (resources) to produce the output. The four arrows further define the functional model into an ICOM (Input, Control, Output, Mechanism) Model showing the necessary resources, constraints, inputs and outputs of the function.

An example of an IDEF0 model, "Implement DoD Data Standardization" is shown as Figure 1.

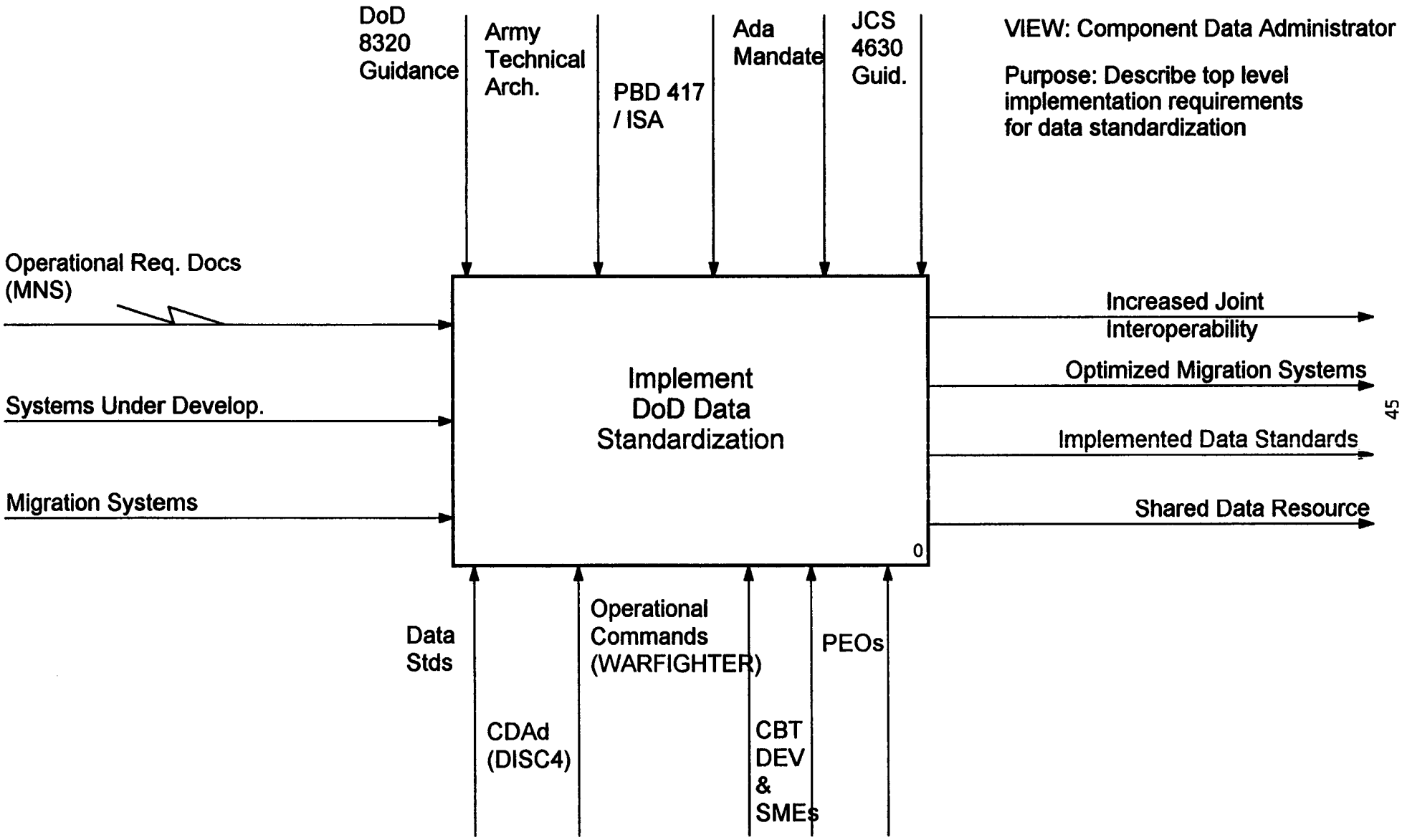
Data Modeling using IDEF1X

Federal Information Processing Standards Publication (FIPS) 184, "Integration Definition for Information Modeling, extended, (IDEF1X)", states that, "Information models produced through the IDEF1X semantic data modeling technique represent the structure and semantics of information within the modeled system or subject area."

The principal objective of IDEF1X is to support integration. The approach is to capture and use semantic definition of data resources referred to as a "Conceptual Schema." This schema provides a single integrated definition of the data within an enterprise which is unbiased toward any single application of data and is independent of how the data is physically stored or accessed. The primary objective of this conceptual schema is to provide a consistent definition of the meanings and interrelationship of data which can be used to integrate, share and manage the integrity of data.

The IDEF1X consists of three key parts: Entities which are things (nouns) of interest to the organization; Attributes are the properties or characteristics common to some or all instances of an entity, where they represent the use of a domain in the context of an entity; and, Relationships which represent the association between two entities or between instances of the same entity.

USED AT:	AUTHOR: Burke, John D.	DATE: 30 Apr. 1995		WORKING	READER	DATE	CONTEXT:
	PROJECT: Implement DoD Data Standards	REV: 20 July 1995		DRAFT			NONE
			■	RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION			



NODE: A-0	TITLE: Implement DoD Data Standardization	NUMBER:

45

Figure 1

An example of a portion of the IDEF1X data model used for the Allied Tactical Command and Control Information System (ATCCIS) which is directly correlated to the C2 Core Data Model is shown as Figure 2.

Limitations on using Activity and Data Modeling

The use of activity models to define requirements and broad enterprise wide information flows is undeniably useful. The conceptual and logical data models are equally valuable in the definition of data and the relationship of data within entities and between entities. Although the relationship between the process and the data models is intuitive, a study commissioned by the Air Force found that, "When defining requirements we are primarily concerned with two elements: data and the use of data. The fundamental flaw of trying to trace these types of requirements from an IDEF model into an object-oriented model is the data (IDEF1X) and the use of data (IDEF0) are not directly integrated."

The DoD, recognizing the structural limitations of the two tools, enforces an environment that provides a common framework for the design and development of process and data models. DoD Procedure 8320.1-M, "Data Administration Procedures" explains that, "Data models and activity models are used as a principal mechanism for managing the data asset, and are aligned to each other through common missions, policies, goals, doctrines, tactics and operations orders."

A second limitation in the use of IDEF1X as a data modeling method is that is has limits for design and implementation directly into software applications. The US Army Electronic Proving Ground Joint Data Base Handbook (DRAFT) explains, "It is important to note the limitations of IDEF1X methodology; it is a tool for modeling the structure of data and their relationships. Its focus is primarily on data and relational modeling and is not an object oriented methodology."

Information Exchange Mechanisms

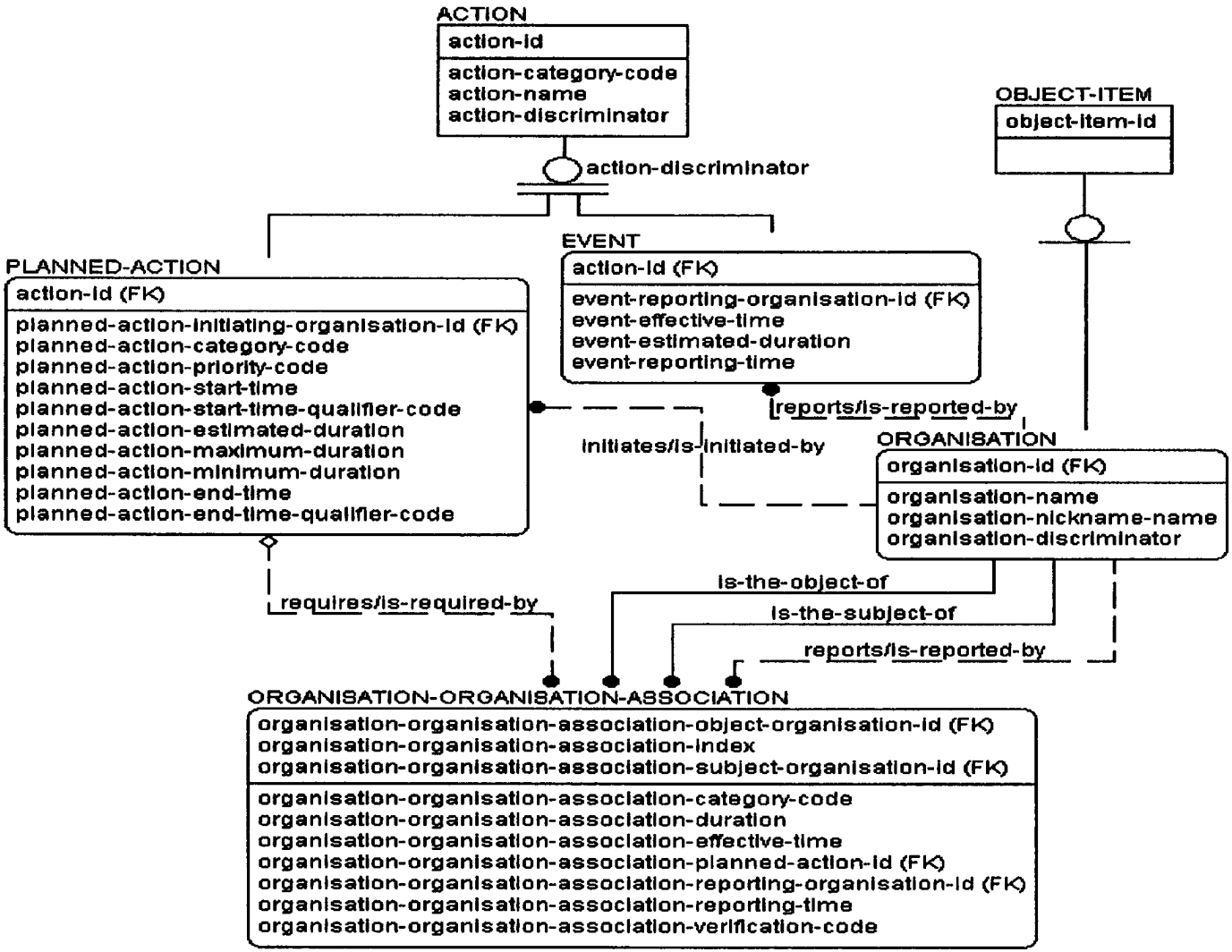
Information exchange represents a very significant requirement for joint interoperability across DoD. The DISA is charged, through DoD 4630.8, to:

1) Identify and maintain a master list of C3I systems and their interoperability requirements.

2) In collaboration with other DoD components, identify requirements for information exchange and develop standardized procedures and formats for information flow among C3I systems.

Information exchange uses the structured means of formatting data in a machine readable form and either a bit or character oriented scheme for transmission across communications media. As part of the modeling and simulation efforts, performance and physical design are a primary consideration. The realities of using narrow-band tactical line-of-sight radios at 16 Kilobits per second through the fiber optic capability of 400 Megabits per second data transfer is necessary if

IDEF1X Data Model



47
Figure 2

we are to achieve the DoD goal of, "Appropriate data available to the warrior in the foxhole and the commander in headquarters, in the type and form needed for the functional process being formed."

Eventually we expect to see the NATO Level 5, database to database, direct interoperability between different computer systems and information sensors and sinks. Today, that is not possible due to a variety of reasons to include operational considerations and the investment in software, training, and joint and combined interoperability. Data transport at the operational and tactical level is done by two principal means: character and bit oriented. The JCS Publication on Joint Interoperability for C3I recognizes the need for standard data and the use of standard databases. Data base standards include the logical structure and the data elements. Unfortunately, present operations will only support computer to computer bulk data transfers including standard formats for initial or replacement data loads and for data base maintenance purposes. These standard formats are the United States Message Text Format (USMTF), a character based messaging system; and the Variable Message Format (VMF) system, a bit oriented messaging system.

Joint Pub 6-04, "US Message Text Formatting Program" provides the management and documentation for character oriented standards. The Joint Pub cites the benefits of character oriented message standards which improve interoperability through: 1) Producing messages that can be read by humans and processed by machine; 2) Reducing time and effort required to draft, transmit, analyze, interpret, and process messages; 3) Provide uniform reporting procedures to be used across the range of military operations; and 4) Facilitating exchange of information between the US and multinational commands.

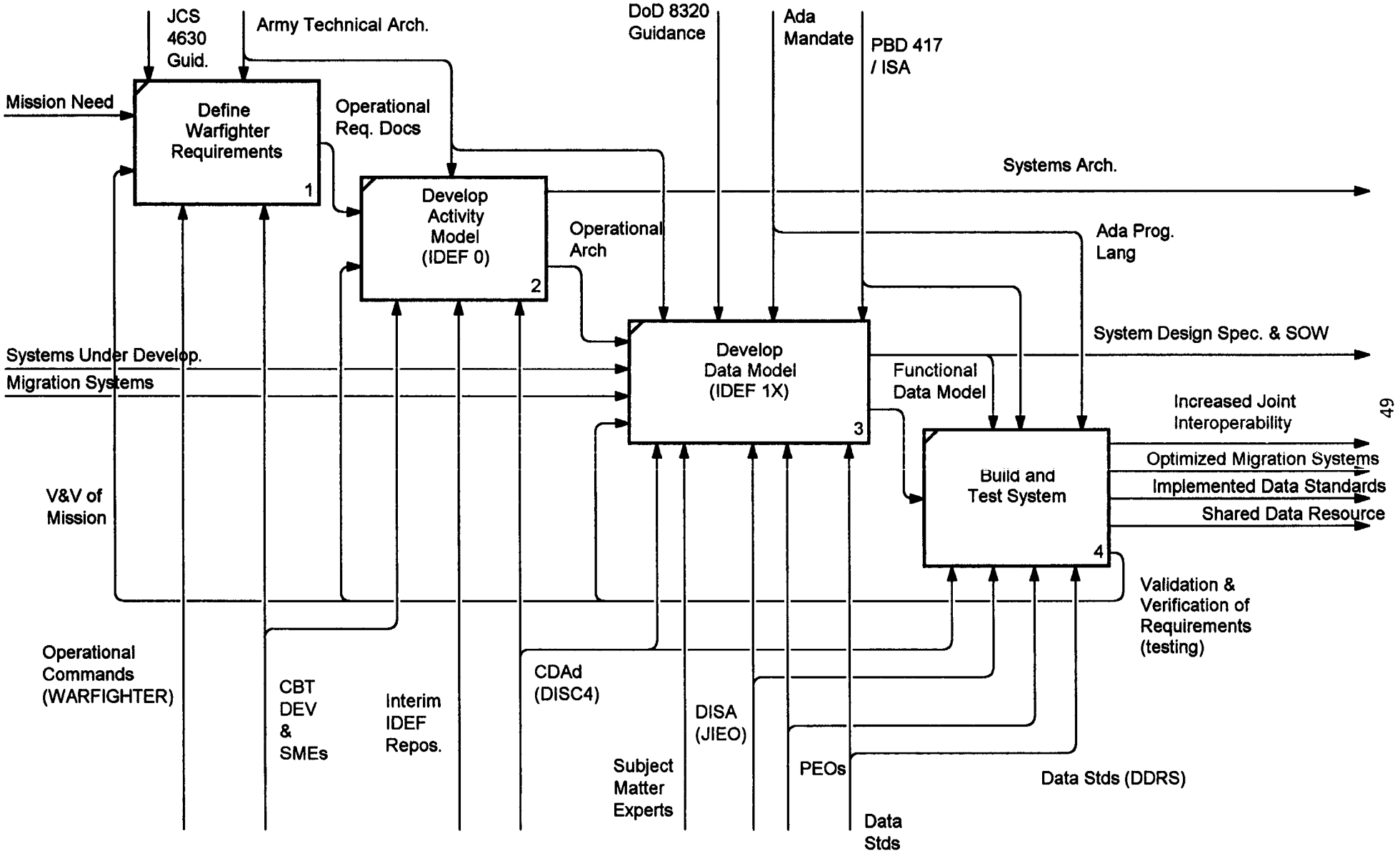
Procedures for bit oriented messages are described in Joint Pub 3-56.2, "Tactical Command and Control Planning Procedures for Joint Operations." Bit oriented message standards provide message formats for data links between command centers, sensor platforms, and weapon platforms. The primary advantage of bit oriented messaging are the efficiencies in data transport and the manipulation of the data by computer processors.

5. JOINT C4I INFORMATION SYSTEMS INTEROPERABILITY

The preceding discussion has outlined the foundation for achieving joint interoperability for information and C2 systems. The remainder of this section provides a recommendation and discussion of how the Army is using this technique for its efforts in the Army Battle Command System (ABCS) which includes the Army portion of the Global Command and Control System (GCCS).

The graphical portrayal of this section is shown as Figure 3, "Implement DoD Data Standardization." These recommendations are not intended to be all inclusive of the complete life cycle system development process; however, they form the basis for a clear understanding of the central thread required to achieve interoperability between information systems.

USED AT:	AUTHOR: Burke, John D.	DATE: 30 Apr. 1995		WORKING	READER DATE	CONTEXT:
	PROJECT: Implement DoD Data Standards	REV: 20 July 1995		DRAFT		TOP
			■	RECOMMENDED		
	NOTES: 1 2 3 4 5 6 7 8 9 10			PUBLICATION		A-0



NODE: A0	TITLE: Implement DoD Data Standardization	NUMBER:

Figure 3

The first step or activity is the clear *definition of warfighter requirements*. These requirements may be found in a variety of documents to include Mission Needs Statements (MNS), Operational Requirements (ORD) documents, system test plans, operations plans and orders, campaign plans, and contingency plans. Additionally, technical documentation from the systems specification and existing and commercial equipment and software may lend insight into the requirements process. Successful requirements documentation requires the participation by the combat and materiel developers, operational commands, and their respective staff and field agencies and commands. The desired output from this step is an agreed upon set of operational requirements with measurable and quantifiable results.

Second, *develop an activity or process model using IDEF0* of sufficient detail that specific entities are identified. Where possible, use the activity models developed by other agencies. The DISA Joint Interoperability and Engineering Organization (JIEO) Center for Software (CFSW) is a source for locating models that have already been done. The Army, under the leadership and direction of the Deputy Chief of Staff for Operations (DCSOPS) is conducting a Service-wide initiative to develop an operational architecture using IDEF0 activity models. The ABCS (Army Battle Command System) has developed a limited scope activity model for Force XXI operations. The output of this step is an operational architecture that defines and describes the information exchange requirements, as well as the information exchange requirements to supplement a systems architecture.

The third step, *develop the data model using IDEF1X*, is predicated on the thorough design of an activity (IDEF0) model. This step requires the building of a conceptual as well as a logical data model. It uses of the Defense Data Repository System (DDRS), managed by DISA/JIEO Center for Software as well as the Personal Computer Access Tool (PCAT). Using the DoD 8320 series guidance for data standardization AND the process models, the data model becomes detailed enough to evaluate and predict performance and quality of design. It involves subject matter experts, system engineers from the program executive offices, software engineers knowledgeable in Ada and operating systems, and constant involvement with the DISA Center for Software for incorporation of the model and standard data elements into one of the functional data models. The output of this step is a fully attributed data model, the System Design Specification, and a portion of the product's Statement of Work.

The fourth step, *build and test the system*, is now based on the structured analysis and design of the activity model and the data models, standard data elements, and the operational requirements documents. As the system undergoes development, it is constantly evaluated for quality of design (validation) and quality of conformance (verification) all the way back to the initial requirements definition. The output of this step is the increased probability of joint interoperability, implemented data standards, and a shared data resource to be reused regardless of the physical location or media.

6. CONCLUSIONS

The evening news, daily paper, command briefings, and budgetary pronouncements reinforce the current military situation of smaller forces, decreased reaction time, and the

likelihood of using an "ad-hoc" force package tailored to the particular requirement. There is little time, if any, to spend on "on-the-fly" engineering of information and C2 system interoperability within services, between services, and even between national military forces. We are forced to specify requirements, model and simulate the probable information exchange, quickly and efficiently standardize our data, and incorporate the engineering into systems. To do otherwise is to be left behind, wanting.

The foundation of joint interoperability is standard data. It is persistent, reliable, accessible, and relatively simple. However, it is only standard data if the entities, elements, and metadata are of sufficient quality to be used directly by systems analysts and software engineers. These analytical investments are represented by the data models which are developed in the context of a known process or activity. Standard data, models, and requirements definition in and of themselves will not ensure joint interoperability; however, they do provide a wealth of information and an implied tool set available to the commander to execute his mission and prosecute the battle.

7. RECOMMENDATIONS FOR FURTHER STUDY

The following represent several areas where I believe further study would improve the utility and understanding of the four step process described above:

Dynamic modeling of process models. As noted earlier, there is not a direct means of linking the activity and data models. Additionally, once the process model has been developed in sufficient detail, i.e., to the entity level, a means should exist to demonstrate the activity using discrete event simulation for validation of the model.

Direct implementation IDEF1X data models into systems design. Although the IDEF1X provides an excellent means of representing the logical structure of the data, it has not been well demonstrated that the data model can be transformed into modular code, especially Ada. Normally, it is assumed that the data model must undergo severe de-normalization in order to reach a modicum of performance.

Object oriented design and IDEF1X models. The DoD appears to be split into two camps: object oriented design and analysis and structured design and analysis (IDEF). Other than the USAF sponsored study cited previously, there is little analysis to draw a conclusion for the system designer/architect to select a methodology.

Defense Data Repository System (DDRS) expansion. The current DDRS and its consumer tool, the PCAT, are currently limited in the representation of standard data (character) and the compatibility of the standard data and other mandates such as Ada and SQL. An area for further research is the incorporation of bit and multi-media (imagery, voice, video) into the central repository. Additionally, a reconciliation between the acceptable representations for standard data and the acceptable data types in Ada and SQL should be accomplished for effective system software development.

LTC John D. Burke is the Chief, Data Management, Headquarters Department of the Army, Office of the Director for Command, Control, Communications and Computers (ODISC4), the Pentagon, Washington, DC. He has served in this capacity since July 1994. His duties include serving as the subject matter expert for Information Standards in the development of the Army Technical Architecture; Joint Command, Control, and Communications Interoperability; Allied Tactical Command and Control; and System Implementation using Data Standardization. LTC Burke received his B.S. in Accounting from the Florida State University and M.S., with honors, from the Air Force Institute of Technology, Systems and Logistics. He is a member of the Institute of Electronics and Electrical Engineers (IEEE) Computer Society and Armed Forces Communications Electronics Association (AFCEA).

LTC John D. Burke
HQDA, ODISC4
ATTN: SAIS-ADO
107 Army Pentagon
Washington, DC 20310
Voice: (703) 695-4553; Fax: (703) 695-5213
Internet: john.d.burke@pentagon-1dms2.army.mil

AFCEA

DOD Database Colloquium
"Emerging Technology for Database Interoperability and Data Administration"

August 28-30, 1995

Abstract

Data: The Critical Enabler for Managing Information Technology Assets in DOD

Authors:

Andrew Verga
Sr. Facilitator
Wizdom Systems, Inc.
1700 Diagonal Road
Alexandria, VA 22314
(703)548-7900 (phone)
(703)548-7902 (fax)
aver@wizdom.com

Betsy Appleby
Telecommunications Specialist
Information Technology Asset Management (ITAM) Project Team Leader
Defense Information Systems Agency
5600 Columbia Pike
Arlington, VA 22041
(703)681-2239 (phone)
(703)681-2781 (fax)
applebyb@cc.ims.disa.mil

"We've got to kill some redundant systems."
Emmett Paige, Jr., Assistant Secretary of Defense, C3I

"The right asset to meet the request at the right time at the right price."
ITAM Project Team

1. INFORMATION TECHNOLOGY ASSET MANAGEMENT (ITAM) PROJECT OVERVIEW

Background

The Information Technology Asset Management (ITAM) project falls under the umbrella of the DOD Corporate Information Management (CIM) initiative. In the DOD, Corporate Information Management is a strategic, collaborative management initiative to guide the evolution of the DOD Enterprise and capture the benefits of the information revolution. It represents a partnership of functional and technical management to achieve a combination of improved business processes and effective application of information technology across the functional areas of the DOD.

Purpose of ITAM

The effective utilization and management of an enterprise's assets is critical for the enterprise's success. When these assets are the means by which communications, coordination, command, and control are executed, their importance to the fulfillment of an enterprise's mission becomes immeasurable. The Department of Defense (DOD) has recognized the crucial role that IT asset management strategies play, their effect on the Department's capabilities to use IT assets when required, and their effect on the Department's capabilities to fulfill its mission. The DOD Information Technology (IT) baseline consists of IT Assets that are managed by DOD Components. Responsibility for IT asset management is shared across the Department. The size and complexity of IT necessitates the use of uniform management disciplines and business concepts to capture, assess, and improve the effectiveness of the management of the IT baseline in support of the DOD mission. A standard ITAM process using evolving support tools was identified as a high-priority need and essential to the management of the IT assets to support mobilization and the goals of the National Performance Review. The ITAM function will provide the Department with heretofore unavailable access to creating and maintaining interoperability between functions and Components.

The overall objective of the Information Technology Asset Management Functional Process Improvement Program is to implement common, standard policies, procedures, and automated systems to better manage the planning, procurement, utilization, maintenance, tracking, and disposal of the DOD's information technology assets leading to fast, effective mobilization, total asset visibility, and interoperability at lower cost. To achieve this objective, the ITAM project had to broaden the definition of IT assets from the conventional one (i.e., only hardware and software) to include hardware, software, data, IT services, documents (plans, blueprints, contracts/warranties, models, etc.), telecommunications equipment, network systems, IT personnel skills inventory, architectures, and new technologies that have a direct relationship to the design, development, maintenance or operation of automated information systems and supporting telecommunications.

Problems with Current ITAM Environment

Currently, the DOD components perform the ITAM functional activity using different policies, procedures, systems, and databases. Problems encountered in this "AS-IS" environment that the ITAM Functional Process Improvement (FPI) Program was charged with addressing include:

- A dynamic and complex IT asset baseline.
- Automated systems that are not interoperable.
- No common discipline for managing IT assets. DOD information technology assets are subject to multiple management disciplines. The processes, institutions, procedures, systems, and databases implementing those disciplines are different according to the respective management practices of the DOD Components.
- Heterogeneous databases. The current DOD listing of information technology assets exists in multiple heterogeneous databases operated throughout the DOD Components. The assets themselves may or may not be described in the databases.
- Non-standard asset descriptions resulting in incorrect management interpretations about that data.
- No means to monitor the IT modernization process.
- Largely inaccessible databases due to DOD Component originator control.
- The use of systems incorporating non-standard data elements and metadata to describe information technology assets.
- Inaccurate, incomplete, and out-of-date data within the database.

Consequently, DOD is unable to manage IT assets effectively to support mobilization and modernization effectively. Access to the IT assets is deficient with respect to overall military mission, business needs, and priorities.

ITAM Functional Process Improvement Project

Recognizing the above problems, the Office of the Secretary of Defense (OSD) has chartered re-engineering initiatives to integrate and standardize the asset management processes that effectively manage DOD IT assets.

Several technical efforts have emerged to alleviate these problems. These efforts include the development of:

a. A DOD technical architecture framework and standards profile information management;

b. Repositories for standard metadata, reusable software components, process and data models, and computer-aided software engineering (CASE) tool sets; and

c. Methodologies and software systems that maintain and provide for manipulation of information technology asset configuration information and data on central design activities, data processing installations, and telecommunication systems.

These concurrent initiatives form only a portion of a coherent, common ITAM discipline. The ITAM program was initiated to provide an appropriate vehicle for defining and administering an

effective ITAM management discipline. This common discipline is necessary to realize the vision for DOD IT in the next decade:

> Make information available to users anywhere at the precise time and location the information is needed and in the format desired, whether the user be on the battlefield or in the office.
>
> Provide users with standard business processes supported by substantially transparent information technology.
>
> Achieve an integrated asset management environment.

ITAM Project Goals

There are five broad goals for ITAM. These goals represent the framework for organizing the major projects and efforts in the ITAM initiative. The goals are summarized below and are further explained in the following sections.

1. Establish ITAM policies and management structure.

2. Re-engineer DOD functional ITAM processes to determine best practices for enhancing ITAM activities.

3. Minimize duplication of IT asset information.

4. Implement a worldwide IT management infrastructure.

5. Integrate ITAM processes to improve consistency within the ITAM function and reveal greater opportunities for increased technical interoperability.

2. REQUIREMENTS FOR TO-BE ITAM FUNCTION

1. **Unified DOD IT Architecture.** Establish a common, unified DOD ITAM process and DOD standard data requirements needed to support ITAM.

2. **Account for assets for IRM managers.** Provide traceability from functional requirements to the IT Assets Plan, including information system architecture and acquired or operational software and hardware. Plans will be on-line, read/copy only.

3. **Respond to OSD reporting requirements.** Provide the structure and grouping of IT assets and their interrelationships.

4. **Be flexible to avoid costly reengineering.** Provide the principles and guidelines that govern IT asset design and evolution over time.

5. **Provide for reuse.** Provide the capability to analyze and review the IT information for future planning and purchases.

6. **Provide for management information.** Provide a capability to develop and implement management strategies and programs to maintain current, accurate, and complete ITAM databases.

3. BENEFITS OF THE TO-BE ITAM FUNCTION

Improved IT asset accountability and total IT asset visibility

To improve IT asset accountability and achieve total IT asset visibility, four areas are addressed:

(a) Asset data,
(b) The use of asset data,
(c) Enterprise guidelines

a) Asset Data

IT assets will include all tangible items that support the concept of Information Technology. These items will include hardware, software, telecommunications, personnel skills, data, network systems, new technologies, and documentation (e.g., plans, blueprints, and contracts). Total IT asset visibility (TAV) will be provided to the desktop level across the DOD, based on a value-added threshold. For example, TAV will be achieved for assets deemed mission critical or for assets that must be tracked to provide for mobilization readiness. For TAV to be achieved, standardized descriptions of assets must be used; that is, asset data must be standardized. ITAM will establish the data elements for managing IT assets. All Components shall adhere to these standard elements. These elements will be controlled through existing standard data element processes; for example, a data dictionary will maintain the standard naming conventions for asset descriptive data and attributes of the data items (e.g., number of alpha-numeric characters). IT asset profiles will exist and will contain the ITAM standard data elements and define the varying levels of information, planning, and control required at Executive, Service, and Field levels. An asset profile may contain such items as acquisition cost, location, age, maintenance information, serial number, Department of Defense Activity Address Code (DODAAC), and history.

b) The Use of Asset Data

The varying levels of detail available for different user categories will not be a means of reducing access to IT asset data. On the contrary, ITAM data and plans will be on-line and available to anyone in DOD who needs it. The user's need will determine the level of detail that is necessary. For example, an executive at the Pentagon performing mobilization readiness analyses will need to know where 3,000 personal computers can be obtained, how they can be reassigned to an Outside Continental United States (OCONUS) site, and what supply-chain logistics can be established Continental United States (CONUS) as opposed to relying on the local economy at the OCONUS site. Field personnel, on the other hand, will need to be aware of the configuration of each personal computer (e.g., how much memory is available on Capt. John Doe's video card) in order to be responsive to user requirements.

On-line data will virtually eliminate the need for data collection and manual data calls. Data on assets will be entered from the initial user's point of requirement and from the asset's acquisition at point of sale and will be managed consistently across the Department. Bar-coding assets and using intelligent scanning interfaces will reduce manual data entry for activities within the assets' life-cycle, increase the automated collection and verification of asset data, and provide near real-time life-cycle histories of all assets.

Executive, Service, and Field personnel will have instant access to asset information, regardless of ownership or leasing. The on-line system will maintain information on assets throughout their life-cycle and provide mechanisms for near real-time updates, changes, and insertions. The on-line information will contain the status of assets (e.g., their configuration, location, upgrades, charge-back codes, and if leased, all details of the lease). Information about individual or groups of assets (i.e., their profiles) will be accessible by a variety of search keys (for example, by specific piece of equipment, by manufacturer, by vendor, by configuration, by contract, by location, by schedule, and by wild cards). Life-cycle controls will be interconnected so that the DOD can operate its assets in a more cost-efficient manner. For example, extended leases can be more costly and less beneficial than outright purchases. An essential element of the life-cycle control interface is the DODAAC, which contains multiple references and addresses for various points of contact within the asset's life-cycle. The on-line system will support a DOD ITA "marketplace", where Components can buy, sell, dispose of, and lease equipment. Capital expenditures will be reduced through re-deployment.

The information maintained on the assets will be designed to promote efficiency in management and maximize use of the assets. Information such as asset usage on assets with limited life-spans (i.e., three years or less) may prove unwarranted and may be eliminated. User needs and assets will be viewed broadly. For example, graphic representation of the assets and their location will be of use to a property book manager (i.e., which desk, which room, which building, etc.). Configuration managers would be concerned with the capabilities of each machine, and its connectivity. Information on these capabilities would facilitate automated configuration management. Graphic representation of the location of assets may also be of value to Executive planners (i.e., 236 personal computers available at Ft. Brag, 488 available at Ft. Rucker, etc.) when planning for mobilization readiness. After the Executive planner has identified the available computers, configuration managers would be concerned with their connectivity and the capabilities to be exploited at their new site, and with ensuring that software and telecommunications equipment would be included in the shipping orders. On-line audits and on-site reports of assets at all locations will provide necessary checks and balances for logistics operations.

c) Enterprise Guidelines

The on-line asset information system will need to satisfy certain business operations requirements of the DOD. The IT asset life-cycle within the new ITAM processes will begin at the point of the user's initial requirement for the asset (point of requirement). When IT assets are managed from initial point of requirement through to disposal, two benefits emerge: (1) user requirements can be analyzed for trends, feedback, effectiveness, expectations, and satisfaction; and (2) the collection of asset descriptive data will begin even before the asset is acquired, which will allow

the asset to be included meaningfully into the inventory once acquired. That is, as soon as an asset is acquired, the profile will be linked into the DOD-available inventory. This will support the concept of a point-of-sale inventory. The point-of-sale inventory will allow the Department to maximize the use of assets and satisfy planning and control mobilization readiness: Both the requirement and the asset will be controlled at their points of entry into the DOD environment. Additionally, bar-coding will enable automated tracking of the asset as it moves from the point of sale to its destination and facilitate the automation of both maintenance information recording and configuration management.

Improved use/reuse of IT assets

To improve the use and reuse of IT assets, changes to the current processes in four areas must be accomplished:

(a) Asset visibility,
(b) Personnel skills as assets,
(c) Automation, and

a) Asset Visibility

One method to improve the use and reuse of assets will be to increase their visibility. If ITA managers can "see" the assets, then they can use them. Items available from vendors (new) and from within DOD (excessed) will be described in sufficient detail for ITA managers to determine if the item meets their requirements. This description will be standardized so that assets can be reused across organizations and Components. IT asset profiles will be designed with increasing information available as the asset is described in greater detail and its capabilities are specified. For example, an ITA manager may be looking for 486s, and may not be concerned with whether they have Ethernet cards or not, because the network managers have a supply of cards and cables to connect all equipment to the Local Area Network (LAN). Another ITA manager may be concerned with exactly how much RAM is contained in the systems, since the computers will be supporting image analysis. Either ITA manager will not be burdened with too much or too little information on the available assets: it will be on-line if needed.

ITA managers will be able to specify configurations, thus improving their ability to manage their inventory and to have automated support for finding excessed assets that satisfy their IT requirements. A configuration will detail the performance expectations, software, and accessory needs of a group of assets (e.g., a logistics workstation). With this configuration defined, an ITA manager can perform automated searches to find the best fit of its requirements in the excess list. Configurations will be definable at DOD-wide and local levels, and they will be modifiable. If specifications are used to build a configuration, then data on how the hardware and software have satisfied those specifications will also be of value in evaluating items for purchase. The use of configurations may also be of value to enforce compliance to a DOD information infrastructure or an IT architecture.

ITA managers will be able to specify and segment their strategic (mission-critical) assets from their nonstrategic assets and prioritize support services for their strategic assets throughout the

life-cycle of the assets.

b) Personnel Skills as Assets

Just as systems are integrated, so too must be the users of the IT assets within DOD. Training standards for users of IT assets will be developed at the DOD level, in compliance with Personnel guidelines. Components will comply with these standards. This will ensure that personnel capabilities are consistent across the DOD. In addition, end-user skills will be managed, tracked, and evaluated. Training costs, history, and computer skills will be tracked to improve training continuously and to allow technical support organizations to better tailor their services to user needs.

c) Automation

For automation to fulfill its promise of support in the new ITAM processes, ITAM systems and services will be fully integrated and will be able to stand on their own. Planning, procurement, and asset tracking will be integrated and managed as a process, not as individual functional activities. All IT documentation, whether on systems, software, contracts, or procurements, will be standardized across the DOD and followed by Components, when applicable. Documentation will be provided on-line. Status of the ITAM process (e.g., procurement, disposal, etc.) will be available on-line. Automated budgeting (development and management reporting) will be available and interfaced with procurement and real-time Enterprise-wide decision support. Capabilities will be provided to perform on-line Preparation for Overseas Movement (POM) budgeting for IT assets. Planning, reporting, and budget status will be streamlined with data integration. On-line trade-off decision support will be available for such decisions as fix vs. dispose. Assets such as software, data, plans, and architecture will be distributed on-line. Configuration management will be automated. Functions for which ITAM requires interfaces (for example, logistics, acquisition, personnel, and data administration) will migrate their automated systems to open systems standards-based tools and will provide interfaces for ITAM electronic access.

Increased Interoperability

To achieve interoperability, the following elements will be present in the ITAM environment:

(a) DOD architecture and enforcement, and
(b) Open systems.

(a) DOD Architecture and Enforcement

The Technical Architecture Framework for Information Management (TAFIM) defines architecture as the disciplined definition of the IT infrastructure required by a business to attain its objectives and achieve a business vision. It is the structure given to information, applications, organizational and technological means _ the groupings of components, their interrelationships, the principles and guidelines governing their design and their evolution over time. The new ITAM process requires an Enterprise-wide architecture. OSD will maintain centralized control

of that architecture. Components will be responsible for compliance with the architecture. IT acquisition will adhere to the architecture. Enterprise-wide standards and methods will be developed for managing technology.

(b) Open Systems

The TAFIM defines open systems as software environments consisting of products and technologies that are designed and implemented in accordance with standards (established and de facto) that are vendor-independent and commonly available. The DOD will use open systems. All outdated systems will be migrated to open systems. DOD will not be tied to any single vendor's technology and the purchase of proprietary systems will be minimized. COTS usage will be maximized for all new mission applications. A high-level and specific exemption will be obtainable.

Increased Standardization

To achieve standardization, Component representatives will comprise the Enterprise governing body and will recommend, review, and comply with IT standards. Government-unique specifications will be minimized when appropriate. Commercial product and industry-standard descriptions will be used for IT assets. Identification of nonstandard configurations and situations will be automated. Training will be standardized to DOD Personnel guidelines. A standardized symbol set system will be used. Decision support for planning and standards management will be improved. Procedures will be institutionalized to facilitate automation; for example, last revision indicators will be mandatory on all documentation, standards, and plans.

Improved upgrade and IT modernization processes

To improve upgrade processes, users and IT asset managers will have access to on-line information that describes the capabilities of an asset to be upgraded. A technology road map will be available with the DOD information architecture to guide upgrade decisions. On-demand market analysis of current and projected asset values will allow IT managers to examine upgrade opportunities. The timing of asset upgrades and replacements will be guided by the technology road map. Hardware acquisitions will be coordinated with technology innovations to maximize long-term savings for the DOD. Both technology and funding plans will be developed consistent with the technology road map. Optimal technology replacement strategies will be developed.

On-line, real-time acquisition research and modernization planning will be supported. Identification and disposal of outdated assets will be facilitated. Costing analyses will also be supported on-line; for example, comparative analyses between total use (depreciation + operation + maintenance) to maintenance cost for replacement and disposal decisions. Capabilities will be provided to analyze life-cycle costs and will include capitalization and depreciation logic, actuals recording, and variance analysis. Also, "what if" analyses will be supported. Real-time market survey of asset salvage value for trade-in, trade-up, and disposal decisions will be a capability of the on-line ITAM system. Analysts will have the capability to notify management of opportunity areas and decision timing. For analysts to become proficient in these types of analyses, training for the ITAM system will be on-line and multimedia-based.

Improved Acquisition of IT Assets

To improve the acquisition of IT assets, both Enterprise procedures and automation support will be provided.

(a) Enterprise Procedures

DOD Enterprise-wide buy agreements should be established; Components should be able to buy from other Component contracts. Through Electronic Shopping Systems, IT products from multiple sources shall be made available; for example, from the GSA schedule, indefinite delivery/indefinite quantity (ID/IQ) contracts, and Component IT partner outlets. These outlets will be within the DOD and will stock inventory items to be delivered to customers within the Department within two weeks of ordering. Also, the outlets should provide support services for fulfilling customer orders from ID/IQ and GSA schedule contracts and the open market.

Improvements will made to warranty management processes. IT asset managers shall be responsible for extended warranty management, maintaining records and proof of warranty, supporting maintenance negotiations, and proactive resolution of warranty and maintenance expirations.

(b) Automation Support

Both purchased and leased asset acquisitions will be supported by the on-line system. Purchase orders will be generated, approved, and tracked electronically. Managers will be supported in validating the technical correctness of configurations ordered. The on-line system will support the interface with vendors for ordering, resolving order discrepancies, reconciling vendor invoices, and capturing IT asset information. Truly paperless ordering, invoicing, remittance, reconciliation, shipping coordination, etc. will be achieved. IT contracts will be advertised DOD-wide on the system. Electronic shopping catalogs will be linked with the GSA schedule, ID/IQ contracts, and the open market. All DOD Components will have on-line access to all IT contracts. When ordering, the IT asset manager will have maximum flexibility to mix criteria for satisfying an order (best price, best delivery time, best vendor evaluation). IT asset managers will be supported by the system with the capability to simulate technical solutions to their IT requests and plan supply-chain logistics for mobilization readiness.

Strategic Management of the DOD IT Baseline to Achieve Interoperability and Manage Changing Technology

Strategic management of the DOD IT Baseline is not configuration management in the conventional sense. Configuration management is a precise process that occurs at local (base and system) levels. Strategic management of the IT Baseline at the DOD Level is targeted at creating and sustaining interoperability between functions and Components. The DOD's effort to migrate toward using open systems is one method of achieving interoperability. Compatibility and connectivity between systems and functions, which through their business operations require information sharing, coordinated processes, and management, must be supported by both

hardware and hosted systems. Thus, two categories of requirements must be satisfied when attempting to develop and sustain interoperability; and the first necessarily must drive the second.

The first category consists of the business requirements of the Department: unimpeded access to information across Components and functions; information protection and security; ensured continuity of operations through emergencies, decreased operations, mobilization, and war. These requirements will dictate the constraints and timing of the second category (i.e., technical requirements). These technical requirements will be satisfied by establishing, adopting, and enforcing an Enterprise-wide architecture; developing, evaluating, implementing, and improving the plans and processes to achieve the objectives of the architecture; and managing these processes so that technological revolutions that can accelerate the achievement of architecture objectives can be seized and incorporated into the IT Baseline.

The new ITAM function assumes the development and management of a DOD IT architecture. From the view of IRM managers, how, when, or who develops and manages the architecture is immaterial. The desire of IRM managers is a standards-based architecture, which guides the population of the IT Baseline. The architecture will guide the identification and possible selection of new technologies in which to invest, the plan for migration and modernization of the Baseline, and the development of contingencies to ensure continuity of operations.

sequent processes. For example, when the excessed equipment/component's profile is updated and designated for redistribution to a particular organization, this would trigger shipping and receiving processes at the depot and within the receiving organization. Bar-coding would prove beneficial for maintaining instantaneous status on the item being redistributed: as it is removed from storage, as it is loaded onto transportation, as it is received, and as it is finally installed. This last procedure would trigger the update of the user's configuration.

If a procurement is executed or if excessed equipment is obtained to satisfy a user's ITA request, the ITA manager needs to be concerned with the user's current equipment. Can it be reused within the organization or declared excess for redeployment and reuse elsewhere within DOD? Again, this will merely be an update of the equipment/component's profile, which will trigger shipping, storage, and depot processes. Criteria will be established for the release of equipment that has been excessed. This release would be a resale or donation.

TO-BE ITAM Function Summary

Information technology assets are hardware, software, telecommunications, personnel skills, data, documentation (plans, blueprints, contracts), network systems, and new technologies. Consistent management of these items across the DOD begins with the management of their descriptive information. Profiles will be constructed using standard data elements and elements that will be standardized for classes of assets (e.g., software, to house the descriptive information necessary to support planning and analysis and to provide end-to-end life-cycle support from the point of the user's initial requirement through acquisition, point-of-sale inventory, utilization, reuse, and disposal). A profile will contain two or more data elements that uniquely describe a managed IT asset. Each profile contains a unique identifier code and at least one piece of descriptive information.

Using the foundation of the profile, automated information systems that provide total asset visibility, electronic shopping capabilities using Electronic Commerce/Electronic Data Interchange (EC/EDI), decision support, and on-line, near real-time analyses will be made available to support Information Technology Asset (ITA) managers as they acquire, locate and manage, sustain, reuse and redistribute, and dispose of IT assets. The system(s) used to track assets and support planning will be based on open systems standards, will be distributed, and will offer many users simultaneous access and usage. The ITAM system(s) will support cost- and effort-cutting measures, such as Enterprise licensing for software, software ownership transfer, asset sharing, and "just-in-time" user requirements, contracts (partner IT outlets), and asset delivery.

IT asset managers will be supported by geographical information systems that provide detailed ITA infrastructure requirements; for example, telecommunications cable requirements (voice, data, and video), design of the cable routes, manhole and duct systems, aerial cable systems, antenna systems, and distribution node buildings. Interfaces to related disciplines, such as standards, personnel, and finance, will be supported by interactive operational linkages with their information systems and processes. Manual intervention for original IT asset data entry will be minimized and data re-keying will be eliminated entirely. The new ITAM functionality will also accommodate technologies such as bar-coding, which can further reduce data-entry requirements and provide checks and balances for asset management.

IT asset users will be supported through the management of their IT skills. ITA managers will be able to plan proactively for continuing education and training that will best support users and their mission requirements. Training will be included with every asset procurement, so that users will have the opportunity to become accomplished at using their assets. Help desks will be available for ITA users and ITA managers for assistance in using assets and the ITAM processes. On-line systems will provide near real-time status of ITAM processes, so that managers and users can track the progress on the processing of their requests and the delivery of their assets. These systems will accommodate prioritization of mission-critical assets and services, so that the most important needs of the users are met first. ITA managers will have access to simulation and prototyping tools, so that the viability of technical solutions can be tested up front instead of on the user's time. Table 1 summarizes the ITAM Critical Enablers that will take the ITAM initiative from concept to implementation.

Table 1. ITAM Critical Enablers

	ITAM Critical Enablers
1	**Profiles and reporting requirements.** Profiles will be used to manage IT asset descriptive information. The profile will be the mechanism by which the asset can be managed throughout its life-cycle.
2	**Electronic shopping for IT assets.** All existing, approved contracts from which organizations can procure assets will be available for review on-line.
3	**Enterprise licenses for software.** Enterprise licenses will be used to reduce costs for underutilized software and allow application standardization to be initiated.
4	**Simultaneous usage of ITAM system.** ITAM system(s) must be available to many users at the same time.
5	**Property accountability through bar-code technology.** Bar-code or similar technology must be implemented to reduce manual data-entry requirements and improve accountability
6	**Simulation/prototype of technical and architecture solutions.** Architectural and technical solutions can be tested and validated before acquisition and implementation.
7	**Decision Support Systems.** Tools need to be provided to asset managers to assist in making decisions.
8	**Fully integrated Geographical Information Systems.** These systems will provide detailed IT asset infrastructure requirements to aid in configuration management.
9	**Point-of-sale inventory.** The inventory provides property accountability at the time the asset is acquired.
10	**Just-in-time.** Essential to the success of ITAM is the elimination of long lead times for forecasting requirements and acquisition.
11	**Training inherent to software packages.** Training needs to be accomplished throughout the various aspects of the environment, including hardware, software, telecommunications, and architecture.
12	**Help desk.** Knowledgeable personnel need to be available when IT asset users need assistance. Users should be able to speak to trained personnel rather than to voice mail.
13	**Asset redistribution, sales, and disposal.** These processes need to be triggered by changes in asset profiles.
14	**Electronic Commerce/Electronic Data Interchange.** EC/EDI will be used to facilitate paperless acquisition, receiving, and invoice resolution.
15	**Total Asset Visibility.** An entire view of the ITAM world should be provided, from initial user requirements, and introduction of the product to DOD, throughout its life-cycle, to ultimate disposal disposition.
16	**On-line market analysis.** This support will minimize individual budgetary expenditures and lower the entire Enterprise cost.
17	**Interactive operational linkages to other disciplines.** Interfaces to logistics, personnel, data administration, standards, and resource managers.

Biographical Sketches of Authors

Andrew Verga

Mr. Verga is a Senior Facilitator with Wizdom Federal. He currently serves as a group facilitator for DoD Functional Process Improvement (FPI) workshops and as an instructor for Activity Based Cost (ABC) analysis and Functional Economic Analysis (FEA) development. In 1992, he co-wrote the DoD FEA Guidebook, an instructional guidebook to assist DoD workshop groups and financial analysts in preparing Functional Economic Analyses.

Mr. Verga has served as an IDEF activity modeling and cost analysis/FEA team leader on numerous DOD FPI projects including the Defense Information Systems Agency Information Technology Asset Management Functional Process Improvement Project, The Marine Corps Maintenance and Supply FPI project, the Joint Logistics Systems Center's (JLSC) Manage Material Business Process Improvement project and the Defense Logistics Agency's Consumable Item Management project.

Mr. Verga holds an MBA from Boston University and a BS in Mechanical Engineering from Polytechnic University.

Betsy Appleby

Ms. Betsy Appleby works for the Defense Information Systems Agency as a Telecommunications Specialist. She is currently Project Manager for DoD-wide Information Technology Asset Management initiatives. Ms. Appleby came to DISA in 1992 from the US Army, where she entered Federal Service in 1989 through the Department of the Army Career Intern Program. Ms. Appleby has received a Master of Arts degree from Shippensburg University in 1988 and a Bachelor of Arts from Immaculata College in 1987.

REFERENCED DOCUMENTS

1. Office of Management and Budget, Executive Office of the President, Circular A-130, "Management of Federal Information Resources", Parts IV and V.

2. DOD Directives and Instructions:

   DODD 8000.1, "Defense Information Management (IM) Program".

   DODI 5000.2, "Defense Acquisition Management Policies and Procedures".

   DODD 7740.1, "DOD Information Resources Management Program".

   DODD 7950.1, "Automated Data Processing Resources Management".

   DOD 8000.x-M, "Defense Automation Resources Management Manual, (DARMM)"

   DODD 4000.19, "Basic Policies and Principles for Interservice, Interdepartmental, and Interagency Support".

   DODD 7740.2, "AIS Strategic Planning".

   DODI 7740.3, "IRM Review Program".

3. Enterprise Architecture Planning, Steven H. Spewak.

4. Client-Server Strategies, A Survival Guide for Corporate Reengineers, David Vaskevitch.

5. "The Open Enterprise", DATAPRO, #1042, June 1993.

6. "UNIX: Its Background and Future", DATAPRO, #1040, April 1992.

7. "Migrating to UNIX and Open Systems", DATAPRO, #1035, November 1992.

8. "Enterprise Computing Systems", DATAPRO, #1015, March 1992

9. "Integration Analysis and Recommendations for Asset Management", Dr. Michael J. Mestrovich, Deputy Director for Enterprise Integration, 1 November 1994.

10. "Reengineering the Corporation: A Manifesto for Business Revolution," Michael Hammer and James Champy, HarperBusiness, A Division of HarperCollins Publishers, New York, NY, 1993.

11. "Every Manager's Guide to Information Technology _ A Glossary of Key Terms and Concepts for Today's Business Leader", Peter G.W. Keen, Harvard Business School Press, Boston, MA, 1991.

12. "Desktop Asset Management Promises Big Payoff", Strategic Planning, Gartner Team, SPA-650-807, 09/26/94.

13. "Management Strategies: PC Cost/Benefit and Payback Analysis," Personal Computing, Gartner Team, R-824-107, 03/24/93.

INCORPORATION OF EXTERNAL DATA STANDARDS INTO THE DOD STANDARDIZATION INITIATIVE

Ann W. Woody -- Defense Information Systems Agency, Center for Software
Neal A. Levene, CSP -- Vector Research, Incorporated
Dave A. Paolicelli -- Vector Research, Incorporated
L. Tobias Klauder -- Vector Research, Incorporated

1. INTRODUCTION

The purpose of this paper is to describe the tasks and procedures used by the Defense Information Systems Agency (DISA), Center for Software, with support from Vector Research, Incorporated as part of the Computer Sciences Corporation's (CSC) Defense Enterprise Integration Services (DEIS) team to adopt external data standards under the DoD data standardization initiative.

The Data Administration program is broadening to incorporate different types of data standards in support of Department requirements. For example, Electronic Data Interchange (EDI) standards, which are specifically for the interchange of data between business partners, will be supported in the Enterprise Data Model as a separate view. This view will then be mapped to the DoD data standards. An additional benefit is that models of X12 transaction sets, which are a by-product of this process, are created. These models can serve as the first step to a physical database design at the EDI Gateways and are useful in the analysis and design of EC/EDI databases. The approach described in this paper leverages the significant standardization efforts conducted by external standards organizations.

This paper provides: (1) background information on DoD guidance and factors that support the adoption of external data standards, (2) descriptions of tasks performed to adopt external standards, (3) mapping of external data standards to functional area data standards, including a description of using the Defense Data Dictionary System (DDDS) as a support tool, and (4) conclusions and directions on additional work to support the acceleration of data standardization through the adoption of external standards.

2. BACKGROUND

DoD 8320.1 series guidance on data administration and data standardization establishes key policy and procedural requirements necessary to support the shareability of data and the integration and interoperability of DoD automated information systems (AISs). One of the key policy requirements includes the use of applicable Federal, National and International data element standards before creating DoD standards. Furthermore, the proliferation of electronic commerce has made external standards vital to the operations of the DoD. Standardization of this data is aimed at including external data standards under the DOD data standardization

initiative and improving the visibility of data interchange standards and their relationship to DoD-approved data standards used in AISs.

DoD data, as described in DoD 8320.1 series guidance, is standardized through a model-based approach. Data models consist of data entities, the relationships between data entities, and data entity attributes. Data elements are the physical representation of data entity attributes. DoD 8320.1-M-1, Data Element Standardization Procedures, defines the naming conventions as well as the formal review and approval process required for data to be approved as standards in DoD.

External data is data which is defined or created by an organization other than DoD and is captured by used by the DoD. Federal, national and international organizations exist which define data standards for the purpose of providing a common format for exchange of information. Each organization has its own set of procedures for development, review, coordination, and approval of standards; for example, FIPS PUB 45, Data Standardization Concepts & Rationale, defines the standardization process for FIPS. Generally speaking, external standards are not derived through a model-based approach such as the DoD program; however, there is an extensive coordination process for these standards prior to approval. In the case of American National Standards Institute (ANSI) X.12, changes to the standard are identified as early as 18 months prior to the implementing release.

The Federal Government develops Implementation Conventions (ICs) for select transaction sets in each new release of X.12. These ICs narrow the purpose of the data structures and permit a more accurate representation of business rules and data definitions. The Federal ICs, which incorporate DoD requirements, are released within several months after the January issue of the new X.12 release. These ICs are critical to the incorporation of X12 data interchange standards into the DoD data standardization effort.

DISA initiated an effort to incorporate external data standards as DoD data standards, which include, but are not limited to: Federal Information Processing Standards (FIPS), Electronic Commerce (EC)/Electronic Data Interchange (EDI) standards, and International Standardization Organization (ISO) standards. This effort supports (1) DoD 8320.1 series guidance as previously described, (2) Department-wide commitment to use external standards described in the Perry Memorandum of 29 June 94, "Specifications and Standards - A New Way Of Doing Business," (3) DoD mandate to accelerate data standardization as specified in the Perry Memorandum of 13 October 93, "Accelerated Implementation of Migration Systems, Data Standards and Process Improvement" and (4) Section 381 of the National Defense Authorization Act of FY95, which requires the optimization of migration systems and application and use of standard data. The purpose of this task is to adopt external data standards, not to change the data standards. This task will improve the visibility of external data standards for use in AIS design and development efforts. Utilization of these standards will also provide the DoD with a means for improving communications with industry partners.

3. APPROACH

This section discusses procedures for adopting and updating external/industry standards as DoD enterprise data. This approach was developed specifically for adopting ANSI ASC X12 and FIPS data. The process was developed based on 8320 series guidance, and it is generically applicable to any external standard.

This process is composed of repeatable procedures and is supported by automated tools. In designing this process, several factors were key considerations.

1) Our goal was to adopt the external standards with no modification. Basic semantic meanings, field lengths, and domains were never changed.

2) The 8320-series guidance describes a procedure that is applicable to external standards. The process described herein complies with that guidance.

3) The DDDS is the authoritative source for all corporate data standards. The goal of all automated tools developed to support this procedure was to facilitate importing the external standards into the DDDS.

4) This process is meant to be both rapid and repeatable.

Figure 3-1 depicts the process. Each step is discussed individually below.

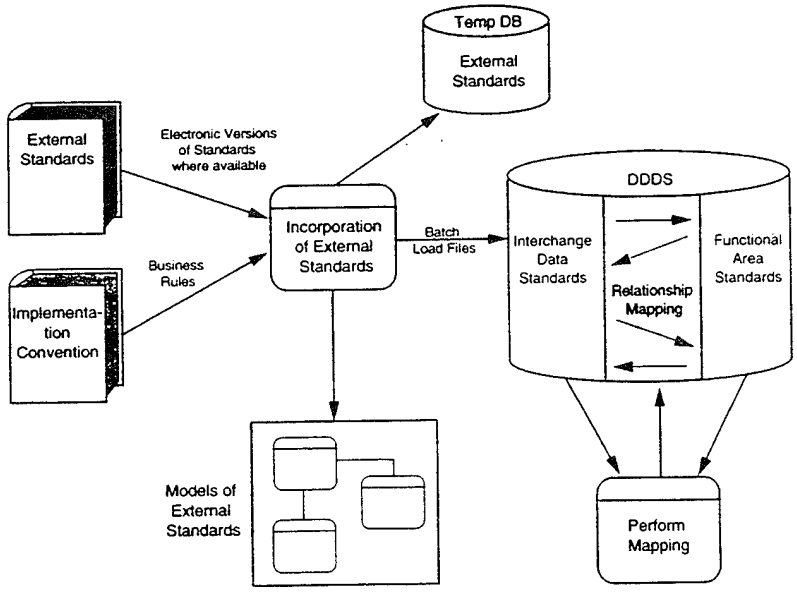


Figure 3-1 Incorporation of External Data Standards Process

3.1 Step 1: Analyze External/Industry Standards

During this step, all necessary materials to adopt an external standard are collected. These materials include: paper and electronic copies of the standard, implementation conventions, data models related to the external standard. A determination is made of the enterprise-level need for an external standard based on the priorities of the Functional Data Administrators (FDAds) and Component Data Administrators (CDAds). Data stewards and key stakeholders for coordination are determined during this step.

3.2 Step 2: Model External Standards

The purpose of this step is to create an operational data model representing the information requirements supporting an external standard. Since most of the external standards that were examined were non-model-based, this stage frequently required significant analysis.

DoD 8320.1 series guidance requires the development of a data model prior to submission of data into the standardization process. Data models are developed to discipline the identification and description of data requirements, and to recognize the context sensitivity of data needs and the business rules that govern the creation and retrieval of data. The first step in modeling external data is to identify high-level entities from the external standard. This may be done by simply identifying entities within the text of the standard, or it may require extensive analysis of the standard including table layouts, Implementation Conventions, if available, and any other information provided. In the case of ANSI X.12, a transaction set data segment (semantic grouping of X.12 data elements) has generally been equated to an entity. Next, relationships between these entities must be derived from the external standard. Implementation Notes often provide business rules within X.12. The model is then further refined by determining cardinality of each relationship; cardinality is determined through analysis of the text provided in a FIPS. The modeling of X.12 is further supported by Federal or DoD ICs as mentioned above. This process places the external/industry standard in the context of the DoD initiative. It does not modify the standard in any way.

During model development, the DDDS must be queried to determine any matches between proposed entities in the external standard model and approved, candidate, or developmental entities in the DDDS. Name matches may not indicate duplicate data. Additional metadata must be compared to determine a complete match. If the name matches but the metadata does not, the name of the external entity in the model should be changed for acceptance in the DDDS and the entity may be proposed as a new developmental/candidate entity. If the entity definitions coincide and the DoD entity is sufficient, the DoD entity should be used. If the external standard is superior for one reason or another, i.e., better meets the information requirement need, the entity should be proposed as a modification to the DoD standard. DoD Functional Area Data Models not yet integrated with the DoD Enterprise Data Model should also be reviewed where possible.

After the comparison has occurred, primary keys and other non-key attributes are defined. As with the identification of entities, the difficulty level of identifying key attributes for an external standard can vary widely. For example, attributes are typically listed in a FIPS with

sufficient explanation to distinguish a unique identifier. On the other hand, the structure of an ANSI ASC X12 standard is complex. Although data elements are listed in X12, primary key roles are not provided, and are difficult to determine. In fact, some entities derived from external standards may not have a unique identifier readily defined within the original structure. Consequently, an artificial key is created to uniquely identify the entity.

Once all components of the fully-attributed model have been established, it is important to inspect and verify that each business rule is maintained.

3.3 Step 3 - Apply DOD Conventions To Data Standards

This task is also required under DoD 8320.1 series guidance. Several conventions are used to adopt external data standards under the DoD initiative. First, following data classification conventions, a prime word and class word are assigned to each attribute in the data model. This classification scheme is used to support the identification and re-use of data items. Second, definition templates for attribute definitions should be used. These templates are used to improve readability of the DoD data dictionary and provide a uniform way to define data concepts. Templates assist in removing guesswork and reducing the time required to adopt DoD data standards. For ANSI ASC X12, which is available electronically, the database is populated automatically with a great majority of the necessary metadata.

3.4 Step 4 - Perform Initial Technical Review

This step is performed to ensure the quality of the data standards. The automated tool generates a Technical Review report for all entities and elements. This process uses the standardized entity and element names, definitions, and domain values created in the previous steps.

The entity and element names and definitions are examined to verify compliance with appropriate DoD 8320-series guidance. The temporary database is modified as required.

3.5 Step 5 - Generate Proposal Packages

The automated tool creates the Entity and Element Proposal Packages, which are the input into the Formal Technical Review process.

3.6 Step 6 - Perform Formal Technical Review

The Government Technical Review is conducted to ensure that the entities and elements contained in the proposal package comply with 8320 series guidance as closely as possible without changing the external standard. The Technical Review also validates that the proposal package data represents the external standard accurately.

3.7 Step 7 - Make Final Modifications

Modifications recommended in the Formal Technical Review are incorporated into the

temporary database.

3.8 Step 8 - Generate Batch Load Files

DDDS import files are created for all proposed data standards. Specifically, DDDS Format 4 is used to import standard data elements. DDDS Format 5 is used to import qualitative domain values. DDDS Format 13 is used to import prime words.

4.0 CROSS-FUNCTIONAL REVIEW

After the external entity and element proposal packages have been imported into the DDDS and placed in "candidate" status by the data steward, they are sent out to the DoD FDAds and CDAds for cross-functional review. The review of external data standards differs from other candidate data standards' reviews. Because external data standards cannot be changed through this process, the review is simply to validate that the candidate proposal package accurately reflects the external standard. In addition, the functional areas should provide a mapping of their data to the external data standards, where applicable.

5.0 MAPPING STANDARDS

Incorporating external data standards, and specifically X12 data interchange standards, into the DoD data standardization initiative supports DoD and the Federal Government's implementation of EDI. The Federal Government has chosen to develop user defined files (UDFs) to support the implementation of EDI. Functional area data is extracted from AISs into these UDFs, which are then translated into the X12 transaction set format. The DDDS provides a means of capturing a mapping between functional area data standards (DoD approved data standards) and the X12 interchange standards.

The DDDS structure will be enhanced to capture the relationship between these standards. In the interim, the Comment Text field of DDDS is used to capture this information.

6.0 CONCLUSION

This paper has described a methodology for adopting external data standards into the DoD Data Administration program. Incorporation of these external standards leverages the extensive data standardization that has occurred outside of the Department. It provides models for non-model-based standards, supporting the first step in developing physical database schemas for archiving data at the EDI Gateway implementation. The ability to map functional area standards to data interchange standards will allow representation of the relationships between data stored within the functional areas and data transferred between trading partners.

AUTHOR BIOGRAPHIES

ANN WALLACE WOODY

Ann Woody is with the Defense Information Systems Agency, Center for Software. She is currently a member of the Data Design Division within the DoD Software Environments Department, working to incorporate external data standards as DoD data standards. Prior to joining DISA, Ms. Woody was with the Defense Logistics Agency, Office of Information Systems and Technology, where she worked with the Logistics community to introduce business process reengineering, Computer Aided Software Engineering (CASE) technologies, and information engineering. Ms. Woody has a B.S. in Management and Marketing from Virginia Polytechnic Institute and State University. She has completed graduate level studies in MIS and systems engineering.

Ann Wallace Woody
DISA/JIEO/CFSW/JEXSH
5600 Columbia Pike
Falls Church, VA 22041
Voice: (703) 681-2507
Fax: (703) 681-2797
e-mail: woodya@cc.ims.disa.mil

NEAL A. LEVENE, CSP

Neal Levene is a Program Scientist with Vector Research, Incorporated (VRI), where he is leading a project to assist the DoD in adopting external data interchange standards into the DoD Enterprise Data Model. Additionally, he provides information resource management experience to other Federal clients, concentrating in the areas of strategic technology planning and data administration. Mr. Levene has an M.S. in Public Management and Policy and a B.S. in Information and Decision Sciences from Carnegie Mellon University. He has received the Certified Systems Professional (CSP) designation from the Institute for Certification of Computer Professionals.

Neal A. Levene, CSP
Vector Research, Incorporated
901 S. Highland Street
Arlington, VA 22204
Voice: (703) 521-5300
Fax: (703) 521-8946
e-mail: levenen@vrinet.com

DAVID A. PAOLICELLI

Mr. David Paolicelli is a Senior Scientist with Vector Research, Inc. (VRI) where he is the current leader of a project supporting the adoption of external data interchange standards into the DoD Enterprise Data Model. Previously, Mr. Paolicelli has provided other Federal clients with support in data administration, information modeling, process modeling, telecommunications guidance, and strategic information technology planning. Mr. Paolicelli has a B.S. in Economics from Radford University.

David A. Paolicelli
Vector Research, Incorporated
901 S. Highland Street
Arlington, VA 22204
Voice: (703) 521-5300
Fax: (703) 521-8946
e-mail: paoliced@vrinet.com

L. TOBIAS KLAUDER

Mr. Klauder is a Scientist with Vector Research, Incorporated, where he assisted with adopting external data interchange standards into the DoD Enterprise Data Model. Mr. Klauder was responsible for overall quality assurance, domain value analysis, and directed the documentation of the data standardization procedures. Mr. Klauder has a BA in Business and Economics from Wheaton College in Wheaton, Illinois.

L. Tobias Klauder
Vector Research, Incorporated.
901 S. Highland Street
Arlington, VA 22204
Voice: (703) 521-5300
Fax: (703) 521-8946
e-mail: klaudert@vrinet.com

Object-Oriented Technology for Integrating Distributed Heterogeneous Database Systems

Dr. Marion G. Ceruti, NCCOSC RDT&E DIV 4221
Dr. Magdi N. Kamel, Naval Postgraduate School
Dr. Bhavani M. Thuraisingham, The MITRE Corporation.

ABSTRACT

In this paper, we describe the application of object technology for heterogeneous database integration. We first discuss the issues of integrating heterogeneous database systems and present a collection of some heterogeneous databases as they occur in the Joint Maritime Command Information System (JMCIS) as an example of a system that could benefit from this technology. Next we describe the need for object technology and its application to handle schema heterogeneity and platform heterogeneity. In particular, we give an overview of the main features of an object-oriented model and why this model is a good candidate as an intermediate model for integrating heterogeneous databases. We discuss the use of Object Management Group's Common Object Request Broker Architecture (CORBA) for resolving platform heterogeneity. Finally, we explore the directions for future research.

Index terms --Object-oriented technology, object data model, relational data model, federated database systems, multidatabase systems, heterogeneous database integration, schema integration, database heterogeneity, semantic heterogeneity, data transformation, command and control systems, object request broker, CORBA

I. INTRODUCTION

The rapid growth of database technologies and networking have had a major impact on the information processing requirements and methods in organizations. Not surprisingly, this topic has received considerable attention in the literature (See for example [2, 3, 7, 8]). In recent years, most large organizations, including the Department of Defense, have seen a dramatic proliferation of incompatible databases and their associated database management systems (DBMS). Sooner or later, these organizations discover the need to integrate the data in these incompatible databases to satisfy their increasing and changing information requirements.

Proceedings of the Twelfth Annual Department of Defense Database Colloquium, "Emerging Technology for Database Interoperability and Data Administration," San Diego, CA, August 28 - 30, 1995.

Various types of heterogeneity have been discussed in the literature (See for example [7, 8]). These include the following:

(i) Schema (or data model) Heterogeneity: Databases in a heterogeneous architecture frequently are represented by different data model with different conceptual schemas.

(ii) Transaction Processing Heterogeneity: Different algorithms for transaction processing are used in different DBMSs. For example, some approaches use locking, others use time stamping, whereas still others use validation mechanisms for concurrency control.

(iii) Query Processing Heterogeneity: Different query processing and optimization strategies may be used in different DBMSs.

(iv) Query Language Heterogeneity: Different DBMSs use different query languages . Even in relational DBMSs, dialects of SQL are used.

(v) Constraint Heterogeneity: Different integrity constraints are enforced in different DBMSs which often lead to inconsistencies. For example, one DBMS could enforce a constraint that all ships must have an overall readiness rating of at least "2" to be deployed, whereas another DBMS may not enforce such a constraint.

(vi) Semantic Heterogeneity: The meaning of data may differ in the different component databases. For example, the port of a ship could mean the home port of origin in one component, whereas in another component, it could mean the port of current location.

(vii) Platform Heterogeneity: Different vendors have produced different hardware platforms, operating systems, network protocols, etc., without considering interoperability issues.

To facilitate interoperability and integration among these heterogeneous databases systems, two approaches, the tightly coupled and loosely coupled approaches [8], have been proposed and developed. In the tightly coupled approach, a unified, global schema is constructed from the underlying individual schemas of the component databases to be integrated. Users and applications need not be concerned with the component database schemas or the integration process. They treat the federated database (FDB) as a single database and issue all query and data-manipulation operations on that schema. In the loosely coupled approach, the component database schemas are not integrated into a global schema. Rather, a method of performing queries on multiple databases is defined and developed to allow access to several or all the component databases simultaneously while hiding the heterogeneity of the underlying databases. A user would be presented with each of the sub-schemas and a powerful data manipulation language or set of tools that enables queries on several or all the component databases. In this approach, the user

needs to know about the structure of the sub-schemas to perform queries and data manipulation successfully. Regardless of the chosen approach, an intermediate representation schema in a semantically rich model is needed to facilitate the transformation of the schemas between the different database systems.

The organization of the paper is as follows. Section II describes an example of an application from a DOD program to illustrate the fact that heterogeneous database integration is a real-world problem. Section II also discusses the need for integrating the component databases in the example. Section III presents an overview of the object-oriented approach and model and discusses why it is a desirable model for facilitating the integration across many heterogeneous database systems as well as administering them. Section IV shows how the object-oriented model can be used as an intermediate data model to address some aspects of heterogeneity, such as schema heterogeneity and platform heterogeneity. It also describes the use of distributed object management system approach to interconnect heterogeneous database systems. Finally, the paper concludes in Section V with a summary and a discussion of directions for future research.

II. A HETEROGENEOUS, COMMAND AND CONTROL DATABASE SYSTEM

The Joint Maritime Command Information System (JMCIS) is an example of a Command and Control system for the Navy, Marine Corps, and Coast Guard resulting from the integration of many legacy systems, including but not limited to the Operations Support System (OSS) and the Navy Tactical Command System-Afloat (NTCS-A). JMCIS was chosen as an example here because many of the major system components and segments were developed separately by different engineers at a variety of agencies, organizations, and commands. Therefore, the aggregate of databases that support these components exhibits considerable heterogeneity, including heterogeneity with respect to platform, query and transaction processing, DBMS, data model, as well as schema heterogeneity and semantic heterogeneity discussed in Section I. For example, the Track Database Manager (TDBM) uses a flat-file data format for real-time applications, and a relational DBMS for analyses and applications with less critical-time constraints. Relational DBMS also are used commonly to manage data in both the afloat (NTCS-A) and the ashore (OSS) variants of JMCIS. The Naval Warfare Tactical Database components, the Naval Intelligence Database (NID) and the Radar Parametric Data Set (RAPADS) are specific examples of data sets managed using RDBMS.

Even within the relational model, DBMS heterogeneity occurs because software from various DBMS vendors has been chosen to support JMCIS database segments of different origins. For example, the JMCIS ashore community has been using Oracle DBMS to manage data from the NID, RAPADS, casualty reports (CASREPS), movement reports (MOVREPS), Naval Status of Forces (NSOF), Status of Readiness and Training (SORTS), employment schedules (EMPSKD), and the

Defense Mapping Agency's (DMA) ports and airfields data sets. These data sets originated in the OSS Integrated Database [4], and have been chosen for migration into the JMCIS ashore variant.

By contrast, JMCIS afloat data are managed using Sybase DBMS for data in the Naval Intelligence Processing System/Services (NIPS) Central Database Server (CDBS). Examples of CDBS component databases are electronic warfare data sets, a message database and a reference database. Moreover, other systems that depend on the JMCIS afloat variant for track and order-of-battle data, also use Sybase DBMS to manage internally generated data for their applications, as well as data derived from JMCIS database components. The DBMS-vendor heterogeneity is not limited to Oracle and Sybase. Informix DBMS software also is included in the Navy's Tactical Advanced Computer-4 (TAC-4) contract and in a recent Coast Guard upgrade.

Some object-oriented technology is being introduced into JMCIS gradually and more object-oriented migration is in the planning stages. New applications are being developed increasingly with a view toward modularity and the open-systems architecture. Similarly, older applications are becoming targets for re-engineering. An effort will be underway to redesign the structure of the Modernized Intelligence Database (MIDB) using an object-oriented data model. MIDB, which originally was developed using a hierarchical data model, is a DOD standard database included in the CDBS of JMCIS afloat. The JMCIS FDB is an example of a federated database system with loose coupling between the components. A goal of the JMCIS integration is to evolve toward tighter coupling. Object-oriented technology can facilitate this tighter coupling.

Not only do the components of the JMCIS FDB exhibit every kind of heterogeneity [3], but any one of many subsets of these components can be present on platform and/or the local area network (LAN) at a given JMCIS site. Some data, such as tracks and casualty reports are dynamic whereas other data sets, such as the DMA ports, are relatively static. Keeping each node current and consistent with all other nodes has been the subject of much effort. Presenting these data in a uniform manner to applications and users also is a challenge because of the wide diversity in data sources. The agency responsible for the production of each data set "owns" these data, and generally offers these data sets to data engineers and developers in a format that is most convenient for the originating agency, rather than the for users. The diversity in the JMCIS FDB is depicted in Figure 1, which suggests the application of an Object-Request Broker (ORB) to address the database heterogeneity problem. The ORB concept is explained in section IV.B.1.

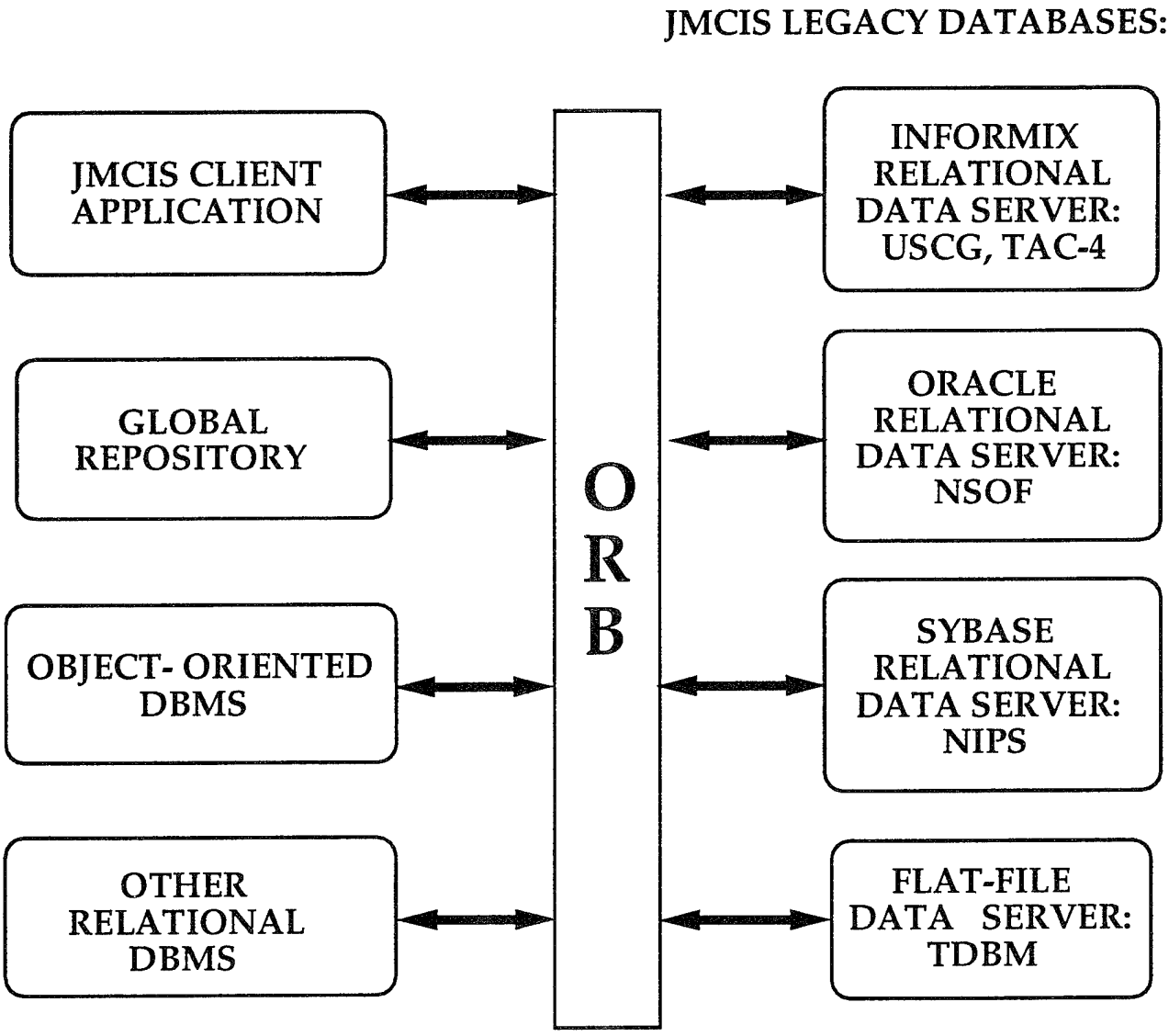


Figure 1. Application of object-oriented architecture to the JMCIS database heterogeneity problem.

III. THE OBJECT-ORIENTED DATA MODEL AS AN APPROACH TO INTEGRATING HETEROGENEOUS DATABASES

In this section we present an overview of the object-oriented data model [1] and a discussion of why an object-oriented data model is particularly useful for integrating distributed heterogeneous databases.

III.A. Object-Oriented Data Model Overview

A fundamental concept in an object-oriented model is the object class, which describes a collection of similar objects, called object instances. An object class has a name, a set of properties that describes its state, and a set of methods that describes its behavior. Each instance of a given class has a value for each property of the class and can invoke the methods associated with the class to perform operations on the properties. Object classes are arranged in a hierarchy known as a class hierarchy. This class hierarchy allows a class, called a subclass, to be defined starting with the definition of its parent class, called a superclass. The subclass inherits the superclass properties and methods, in addition to having its own specific properties and methods.

Whereas a widely accepted definition of what constitute an object-oriented data model has yet to be developed, there is a general consensus on certain core concepts, which are discussed below (See for example [6, 9]). The object-oriented data model is illustrated in Figure 2.

(i) Encapsulation: This refers to the fact that an object has a public interface part and an implementation part that is encapsulated or hidden from public view. The interface part specifies a set of operations that other objects or applications are allowed to perform on the object. The implementation part describes how each operation is used. Encapsulation provides a form of "logical data independence" because one can change the implementation of a method without changing the programs using that method, thus providing for a greater degree of modularity than some other data models.

(ii) Message Communication: Objects communicate with each other through messages. A message consists of the name of an object followed by the name of a method to be executed and optionally any parameters that the method requires for its execution.

(iii) Polymorphism: Polymorphism is the ability of different kind of objects to respond differently to the same operation depending on the type of object.

(iv) Inheritance: A subclass inherits the properties and methods of the superclass, which has two advantages. It allows for a more concise and precise way to model the world and it facilitates the identification of shared specifications and implementations in applications.

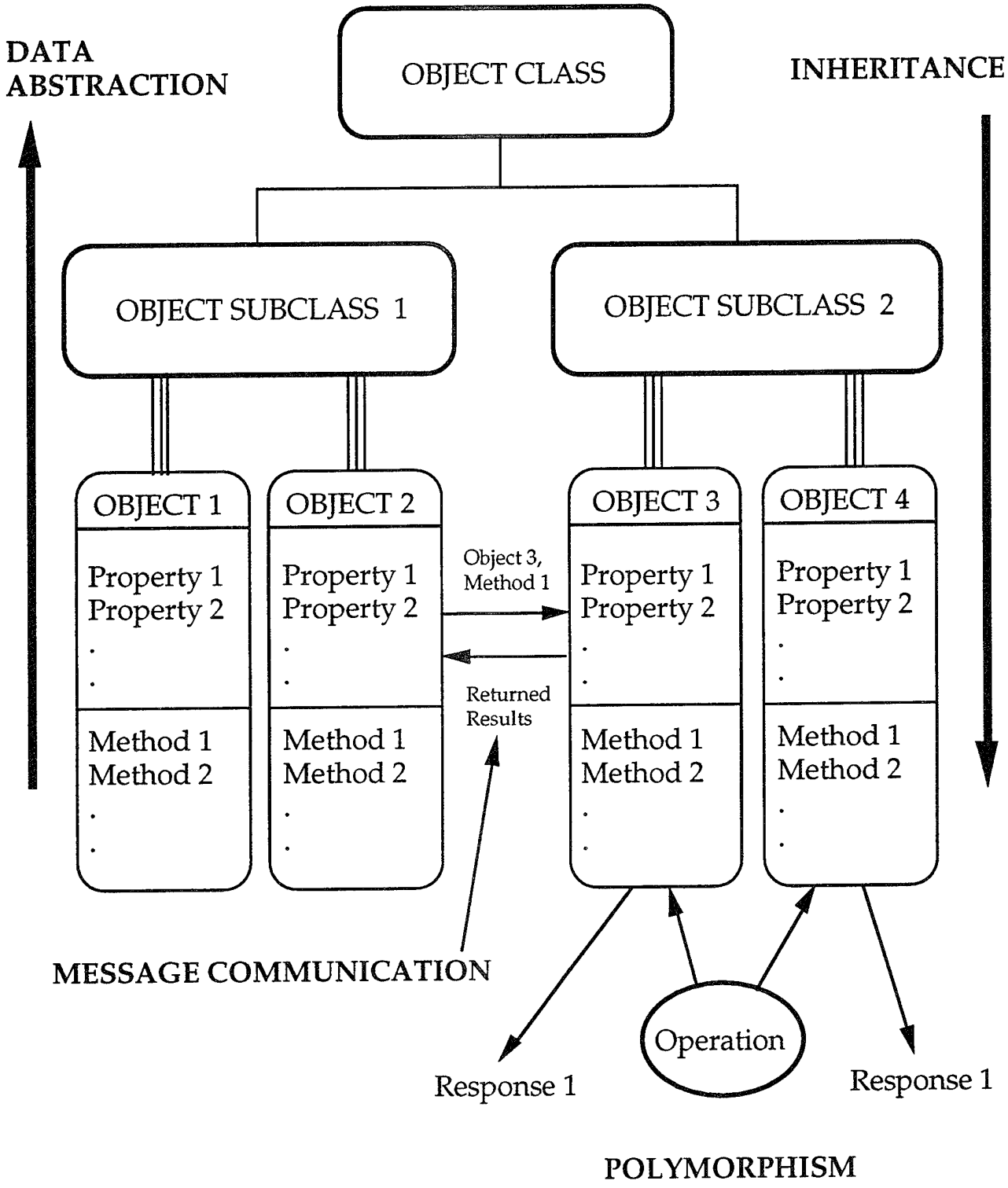


Figure 2. The object-oriented data model

(v) Data Abstraction: The ability to define new high-level data types by combining existing data types to achieve a better representation of the semantics of a new application is called "data abstraction."

In Section IV, the object-oriented concepts above will be related to those of the relational model with a view toward facilitating a data representation transformation between these models.

III.B. Why use an Object-Oriented Data Model for Integrating Heterogeneous Databases?

The object-oriented data model is a particularly powerful model for integrating heterogeneous database systems for several reasons (See for example [2, 10]). First, the extensible data typing feature of an object-oriented model makes the model semantically rich and therefore, a good candidate for an intermediate data model to represent the semantics of any existing data model of the component databases. The use of the object-oriented model as an intermediate data model is illustrated in Section IV.A. Secondly, an object-oriented data model supports both operational mapping and structural mapping. Operational mapping allows the database integration engineer to define the correspondence between operations at different levels of the system that is being integrated. This feature is particularly useful for integrating non-traditional databases that have no formal schema. Third, the inheritance characteristic of object-oriented models simplifies the definition of tailored integrated views, since it allows a view to use existing mappings defined for other views. In addition the object-oriented model could be used to represent the entire distributed heterogeneous system. With this approach, the different components of the distributed heterogeneous system are represented as objects that can interact with each other by exchanging messages. The distributed object management approach is illustrated in Section IV.B.

IV. OBJECT TECHNOLOGY FOR HETEROGENEOUS DATABASE INTEGRATION

In section IV.A we discuss the application of object technology to handle schema heterogeneity (See for example [10]). In section IV.B we discuss how object technology could be used to handle platform heterogeneity (See for example [11]).

IV.A. Object Technology for Schema Integration

Most users find increasing difficult in learning a variety of data representation schemes on systems that they must access remotely. A much more desirable alternative is for the users to know at most one or two schemes to be effective in their work. For example, a user who needs to access a relational database at one site, a hierarchical database at another site, and an object-oriented database at still another site, will be able to access these databases more efficiently using his or her preferred

data model and language. Thus, schema integration will be reduced to transforming the constructs of one data model into those of the others.

In this method, translators can be installed to perform the necessary transformations between the various data representations. This is advantageous because it relieves the users of the task of learning the details of more than one data representation scheme. Unfortunately, this method suffers from a disadvantage; if N different data models are utilized, (N^2 - N) translators will be required. Thus, the task of producing the required translators becomes costly and cumbersome, especially if the number of data models in the system continues to grow.

A more efficient manner in which to interconnect data models is to use an intermediate representation scheme that can act as a central point of translation. The individual data representation schemes are all translated into the intermediate representation scheme. The intermediate representation schemes then are translated back into the individual data representation schemes to allow for a two-way conversion between the individual and the intermediate representation schemes. Thus, for N different data models, only 2N translators are required because in this method, each intermediate representation scheme communicates only with the intermediate representation scheme.

An object-oriented representation scheme was proposed to serve as the intermediate representation scheme [10]. In the present work, we show how an expanded version of this scheme can be applied to an example derived from the SHIP_MASTER_LIST table in the JMCIS FDB. This object-oriented representation is sufficiently powerful to model structural in addition to behavioral properties of entities. Moreover, the majority of constructs of other data models, such as the relational model, can be transformed into constructs of object-oriented data models. For example, schema integration can be accomplished using an object-oriented approach in heterogeneous environment that consists of relational DBMS and a flat-file data server.

Because relational DBMSs are in such widespread use throughout the DOD and elsewhere, we examine the relationship between a relational model and an object-oriented model. In the strict theoretical sense, several steps are required to generate relations from an object-oriented data model. The theory associated with these steps is described in more detail in Appendix A. An intermediate relational data model is generated to be a generic relational data model. The end result is that the constructs of the generic relational data model are transformed into those of a specific relational data model such as one used in commercial DBMSs. The essential concepts are illustrated below with command and control examples.

Object classes are mapped into one or more domains, which can contribute to one or more relation variables (See Appendix A). Relations are then constructed using the relation variables as structures which are filled with values from the appropriate domains. Consider the class SHIP illustrated in Figure 3. This class has

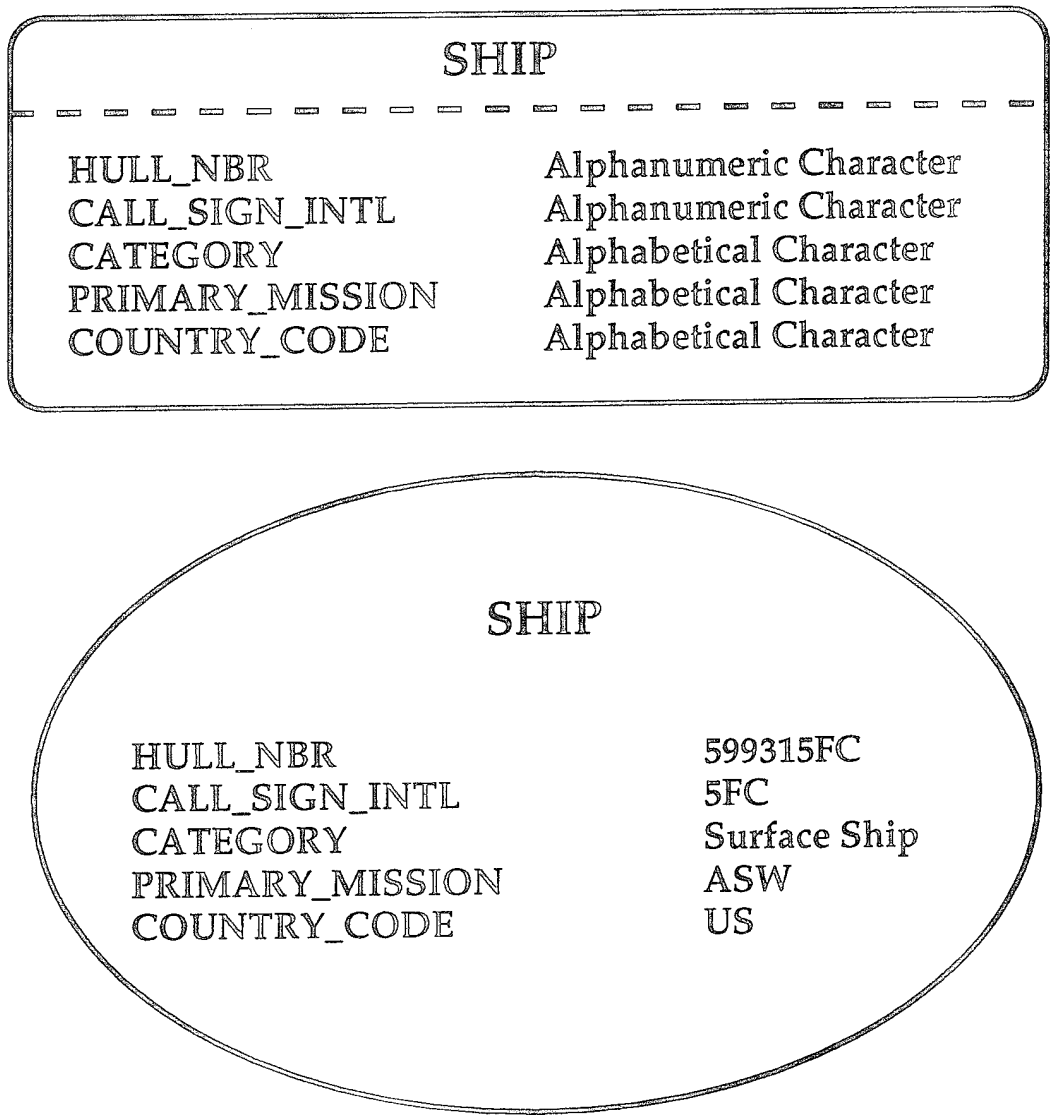


Figure 3. Attributes, data types, and values

attributes HULL_NBR, CALL_SIGN_INTL, CATEGORY, PRIMARY_MISSION, and COUNTRY. It has an instance object whose HULL_NBR is "599315FC," CALL_SIGN_INTL is "5FC," CATEGORY is "Surface Ship," PRIMARY_MISSION is "ASW," and COUNTRY is "US." This ship class is transformed into a complex domain, D_A, consisting the following domains, or data types:

HULL_NBR	set of all alphanumeric vessel identifiers
CALL_SIGN_INTL	set of all alphanumeric international call signs
CATEGORY	set of characters indicating surface ship, submarine, amphibious craft, etc.

PRIMARY_MISSION	set of three-alphabetical characters designating the primary warfare mission area
COUNTRY	set of two-alphabetical characters designating the country that is using the ship

From this complex domain, a SHIP relation can be formed. The object identifiers of the instances can be used as primary keys or a mapping could be provided between these object identifiers and the user-generated primary keys if users need to access the relations directly.

This methodology provides a way to transform constructs such as associations and inheritance. The association between object classes, "SHIP" and "PORT" is "SAILS TO." Associations between object classes are similar to message communications between objects. For example, a particular ship, an object instance of class "SHIP," communicates with its destination (another object instance of class, "PORT," by sailing to it. The association also has attributes PORT_ID, PORT_NAME, HARBOR_TYPE, PILOTAGE, LATITUDE, LONGITUDE, and COUNTRY [12]. A relation can be formed to describe this association by drawing from the domains that constitute the essential object classes. The attributes of this relation consist of the attributes of the association and the identifiers of the classes involved in the association.

Object classes can be arranged into a class hierarchy. That is, VEHICLE class has SHIP as its subclass. Similarly, SHIP has SURFACE SHIP and SUBMARINE as object subclasses. A domain exists for each class and subclass. Depending on the degree of normalization, the relation variable generated for the subclass may contain only the attributes that distinguish the subclass from other subclasses or it could include additional attributes also found in the superclass. Any instance of the subclass SHIP will be represented by two tuples; one in the relation VEHICLE and one in the relation SHIP.

IV.B Distributed Object Management Approach

This section describes how distributed object management systems such as the Common Object Request broker Architecture could be used for heterogeneous database integration.

IV.B.1 The Common Object Request Broker Architecture. The information on CORBA discussed here is from [5]. CORBA consists primarily of the object model, the Object Request Broker (ORB) and object adapters, and the Interface Definition Language (CORBA-IDL). Each component is discussed below.

Object semantics and object implementation are described by the object model. Object semantics include the semantics of an object, type, requests, object creation and destruction, interfaces, operations, and attributes. Object implementation includes

the execution model and the construction model. In general, the essential constructs of most object models can be found in the object model of CORBA.

An essential feature of the ORB is that it enables communication between a client and a server object. A client invokes an operation on the object and the object implementation consists of the code and data needed to implement the object [5]. The ORB provides the required mechanisms to identify the object implementation for a particular request and enables the object implementation to receive the request. The ORB also provides the communication mechanisms needed to deliver the request. Furthermore, the ORB supports the activation and deactivation of objects and their implementations. The ORB generates and interprets object references. To summarize, the ORB provides the mechanisms to locate the object and communicate the client's request to the object. The client does not need to know the exact location of the object or the details of its implementation. Objects use object adapters to access the services that the ORB provides.

IDL is a declarative language that describes the interfaces that the object implementations provide and that the client objects call. It should be noted that the clients and object implementations are not written in IDL. The IDL grammar is a subset of ANSI C++ with additional constructs to support the operation invocation mechanism. An IDL binding to the C language has been specified, whereas other language bindings are in progress. IDL is used to communicate between a client and a server in the following manner. Two types of modules, the IDL stub and the IDL skeleton, are connected to the ORB core. The client's request is passed to the ORB core via an IDL stub, and an IDL skeleton delivers the request to the server object from the ORB core.

IV.B.2. The Use of CORBA for Integrating Heterogeneous Database Systems. Section I presented and overview of various types of heterogeneity and section IV.B.1 explained OMG's CORBA. This section provides a description of some initial directions on using CORBA for integrating heterogeneous databases.

A major motivation for adopting a CORBA-like approach to the integration of heterogeneous databases is the complexity of migrating legacy databases to new-generation architectures. Whereas the migration of such databases and applications to the client-server architectures is desirable, the costs of such migration can be enormous. Therefore, a better approach is to keep the legacy databases and applications and develop mechanisms to integrate them with new systems. These mechanisms include the distributed object management system approach in general and the CORBA approach in particular.

The major advantage of the CORBA approach is the ability to encapsulate legacy database systems and databases as objects while eliminating the need for major modifications as is depicted in Figure 4. However, the techniques to handle the various types of heterogeneity are still needed. This is because the CORBA approach does not handle some problems like transaction heterogeneity and semantic

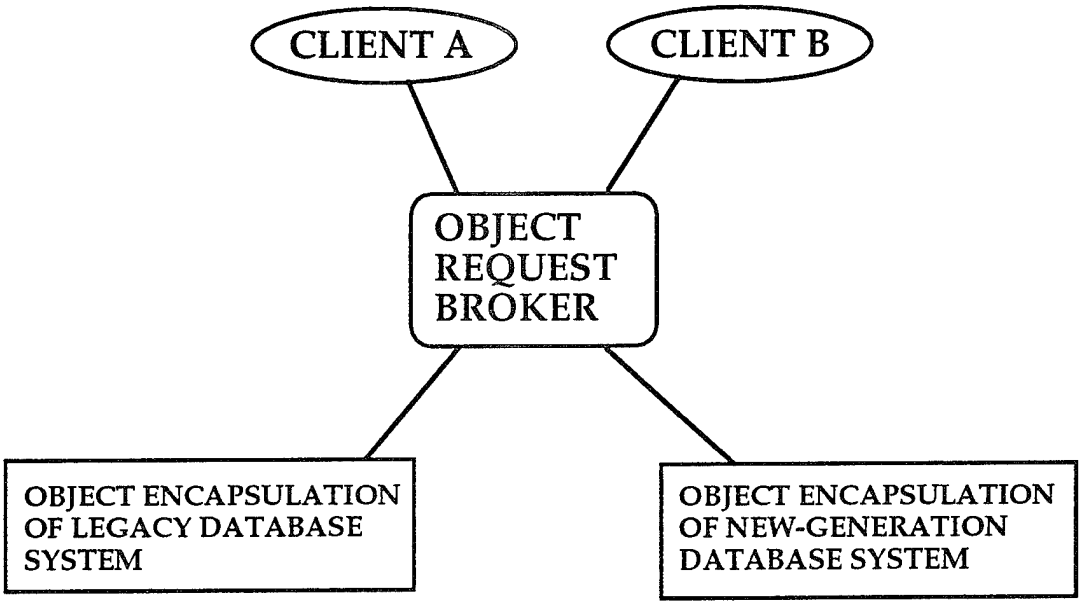


Figure 4. Encapsulating legacy and new-generation database systems

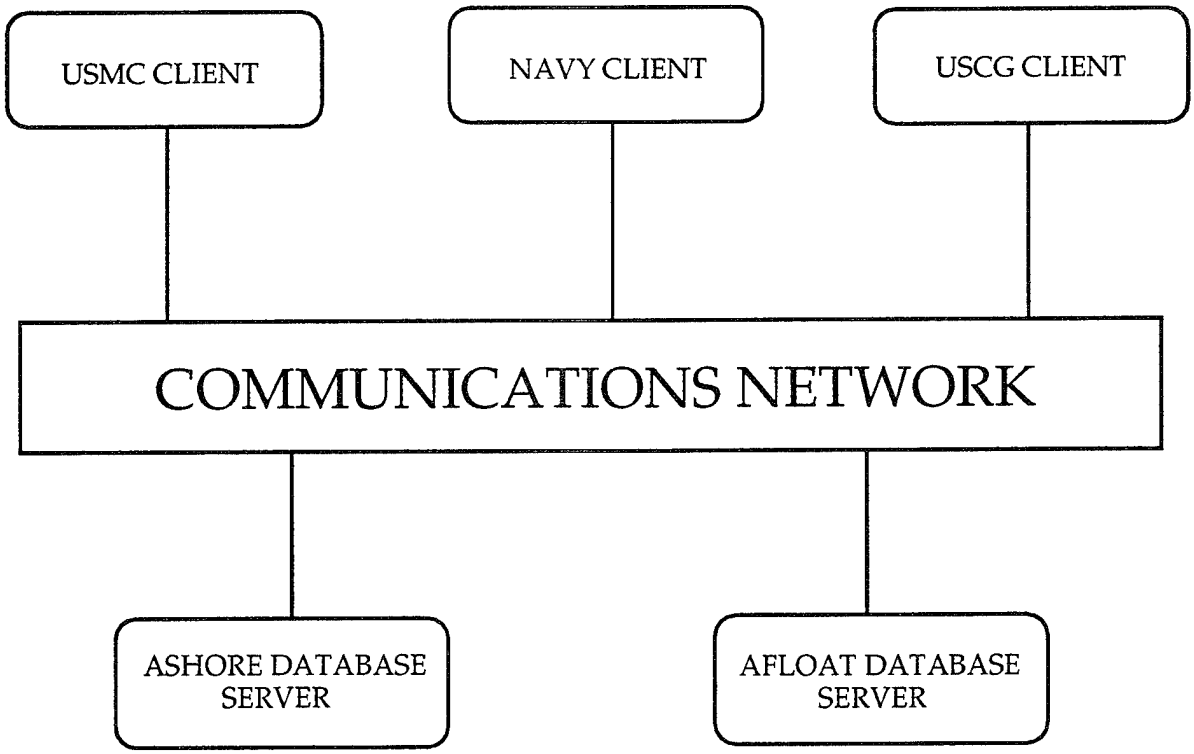


Figure 5. A client-server architecture for JMCIS

heterogeneity. However, the procedures for handling the various types of heterogeneity can be encapsulated in the CORBA environment and invoked appropriately. These concepts are illustrated below with some examples.

Consider the need for clients to communicate with a group of database servers. This is shown in Figure 5, which uses some components of JMCIS as an example. One way is to encapsulate the database servers as objects and have the clients issue appropriate requests and access the servers through an ORB. If the SQL-based servers are used, the entire SQL query or update request could be embedded in the message. When the method associated with the server object gets the message, the method can extract the SQL request and pass it to the server for execution. The results from the server objects are encoded as a message and passed back to the client via the ORB. This approach is illustrated in Figure 6, which contains a generalization of some features of Figure 1 as both include an ORB as the intermediate between clients and servers.

Next, consider the issue of how to deal with a particular type of heterogeneity. Suppose a SQL-based client is present with a server is some legacy database system based on the network model. Then the client's SQL query will need to be transformed into an appropriate language that the server can understand. In Section IV.A, the issues of transforming one representation scheme into another were discussed. The client's request is sent to the module responsible for performing the transformations. This module, called the "transformer," could be encapsulated as an object. The client's SQL request is sent to the transformer, which converts the request into a format that the server can understand. The transformed request is sent to the server object. Note that the transformer could transform the SQL representation directly into a network representation or it could use an intermediate representation to complete the transformation.

The Distributed Processor, which is a module that can perform the distributed data management functions, is responsible for handling functions such as global query optimization, and global transaction management. This module also is encapsulated as an object and processes the global requests and responses. The server assembles the response sent to the transformer to convert into a representation that the client can understand.

If semantic heterogeneity is an issue that requires attention, one may need to maintain a data repository to store the different names given to a single object or the different objects represented by a single name. The repository could be encapsulated as an object which would resolve semantic heterogeneity. For example, referring to Figure 1, a JMCIS client application could request an object to be retrieved from multiple servers. The request is first sent to the repository which will issue multiple requests to the appropriate servers depending on the names used to specify the object. The response also could be sent to the repository so that it can be presented to the client in an appropriate manner. Note that the repository could be an extension of the transformer. All the communications are carried out through the ORB.

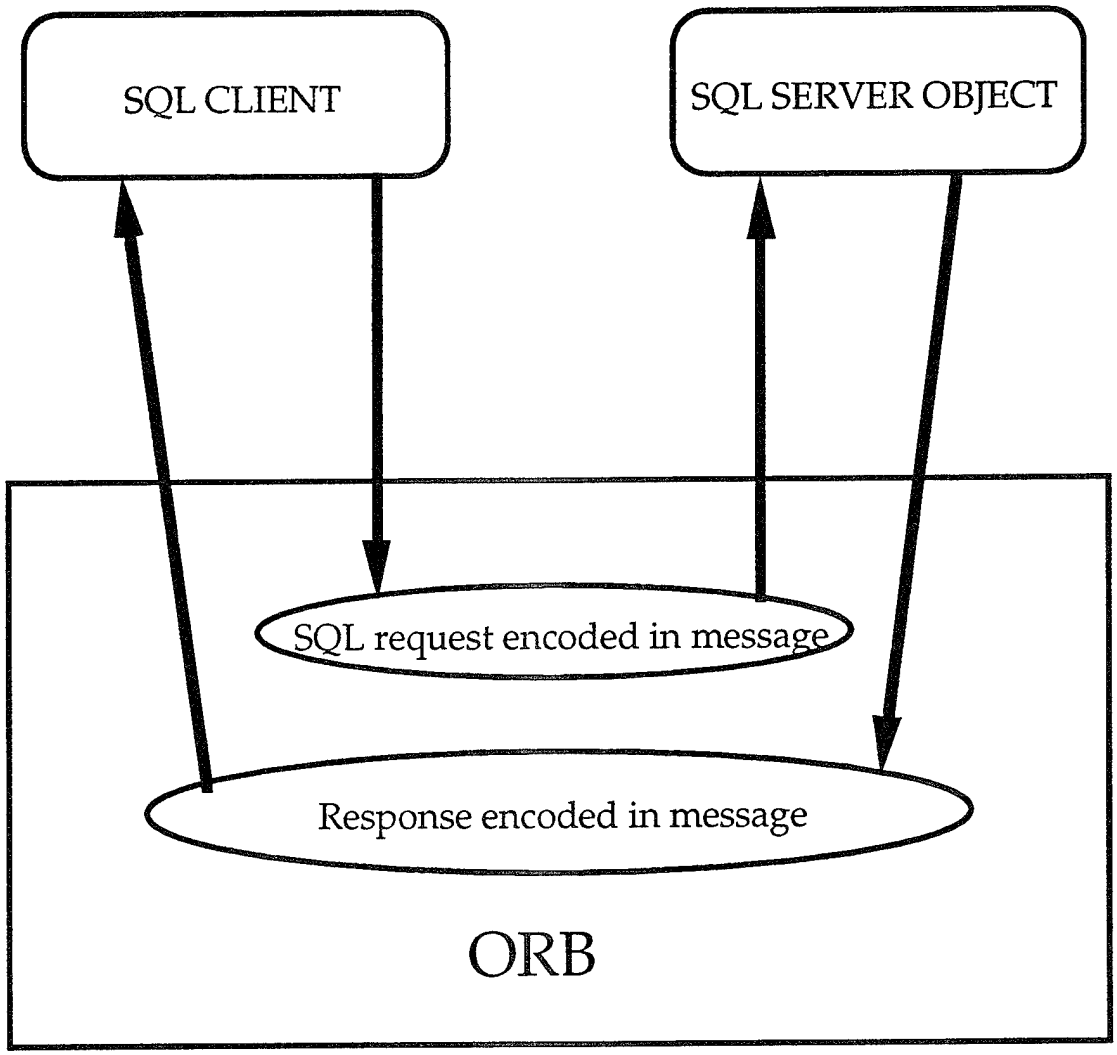


Figure 6. How interoperability is accomplished through CORBA

V. SUMMARY AND FUTURE CONSIDERATIONS

In this paper, we first described the issues involved in integrating heterogeneous database systems, including some real-world examples. We also described the reasons for an object-oriented approach to integrating heterogeneous database systems. We introduced a multi-phased approach that involves several steps between the object-oriented data model, and the relational model. Two types of applications we discussed include the use of the object model for schema integration to handle schema heterogeneity, and the use of a distributed object management system to handle platform heterogeneity.

Several issues need further consideration with respect to the use of CORBA for integrating heterogeneous databases. For example, what is the best level of granularity at which to apply encapsulation? Should a server be encapsulated as an object as described in this paper? How can databases be encapsulated? Should an entire database be encapsulated as an object or should it consist of multiple objects? Should stored procedures be encapsulated also? Although much work still needs to be done, the various approaches that are being proposed to handle these issues are showing substantial promise. Furthermore, until efficient approaches are developed to migrate the legacy databases and applications to client-server based architectures, approaches like CORBA and other distributed object management systems for integrating heterogeneous databases and systems are needed.

REFERENCES

[1] J. Banerjee et al., "Data Model Issues for Object-oriented Applications," *ACM Transactions on Office Information Systems*, Vol. 5, No. 1, 1987.

[2] E. Bertino, "Integration of Heterogeneous Data Repositories by Using Object-Oriented Views," *First International Workshop on Interoperability in Multidatabase Systems*, Kyoto, Japan, April 7-9,1991.

[3] M. G. Ceruti and M. N. Kamel, "Semantic Heterogeneity in Database and Data Dictionary Integration for Command and Control Systems," *Proceedings of the 11th Annual Department of Defense Database Colloquium*, pp. 65 - 89, August 29-31, 1994.

[4] M. G. Ceruti and S. D. Rotter, "Software Reuse Key Element of Navy Data Base Structure," *Signal*, Vol. 48, No. 1, pp. 55-56, September 1993.

[5] *The Common Object Request Broker Architecture*, Published by OMG and X/Open, Addison Wesley, 1992.

[6] M. Loomis, *Object Databases: The Essentials*, Addisson-Wesley, 1995.

[7] P. Scheuermann et al, "Report on the Workshop on Heterogeneous Database Systems," *ACM SIGMOD Record*, Vol. 19, No. 4, 1990.

[8] A. P. Sheth and J. A. Larson, "Federated Database Systems for Managing Distributed Heterogeneous. and Autonomous Databases," *ACM Computing Surveys*, Vol. 22, No. 3, pp. 183-236, September 1990.

[9] D. Taylor, *Object-Oriented Technology: A Manager's Guide*, Servio Corporation, 1990.

[10] B. M. Thuraisingham, "Distributed Object-Oriented, Interoperable, Heterogeneous Database Systems," *Handbook of Local Area Management*, Second edition, Auerbach Publishers, Ed: J. P. Slone, pp. 477-487, 1994.

[11] B. M. Thuraisingham, "Object Management System Approach to Integrating Heterogeneous Database Systems," Submitted to *Handbook of Local Area Management*, Third edition, Auerbach Publishers, 1995.

[12] *World Port Index*, Defense Mapping Agency Hydrographic/Topographic Center, Pub. 150, Fourteenth edition, 1994.

APPENDIX A

OBJECT CLASSES AND DOMAINS: THE INTERMEDIATE STEPS BETWEEN THE OBJECT-ORIENTED AND RELATIONAL DATA MODELS

Object classes in the object-oriented data model map to domains in the relational model. By contrast, object classes do not map directly to either relations or relation variables. Domains, in addition to being conceptual pools of values from which various attributes draw their actual values, are really scalar data types [A3], sometimes known as "atomic," "basic," or "primitive" data types, which also allows for the possibility of strong data typing. These atomic data types can be aggregated into structured data types that are no longer scalar, by means of type constructors. Examples of type constructors include lists, arrays, and relations, in which each tuple has a composite data type that results from the contribution from the scalar data type, or domain, of each attribute in the relation [A1, A2].

Relations, which contain tuples, should not be confused with relation variables, which have no tuples but serve as headers for tuples. A relation variable is a set of ordered pairs (A,V) where A is an attribute and V is its domain. This set serves as the specified heading, H, whose permitted values are relations [A2]. In essence, a relation variable or relvar, R, is a created table structure or heading, whereas a relation, r, constitutes the data fill that is arranged according to that structure [A1, A2]. More explicitly, consider the following expressions, in which relvar, R is a set of ordered pairs:

(1) $$R = \{ (A_1, V_1) , (A_2, V_2) \ldots (A_n, V_n) \}$$

and relation, r is a set of ordered triples:

(2) $$r = \{ (A_1, V_1, v_1) , (A_2, V_2, v_2) \ldots (A_n, V_n, v_n) \}$$

where each A is an attribute and V is the domain or data type that applies to that attribute, and v is the specific value from domain V that is assigned to attribute A in

relation r. Note that relvar, R does not have specific data value assignments, v. The relations, r are simply values of variable, R.

The main focus of this phase of the approach emphasizes importance of domains and conceptually distinguishes between relations (with fill) and relation variables (with structure only) [A1-A3]. However, a mapping procedure that results in the construction of relations is still needed. This is complicated by the fact that the object-oriented model and the relational model are really very different paradigms that do not always have a clear one-to-one mapping. Further complications arise because information contained in the implementation part of an object, which is hidden from public view, may need to be expressed explicitly in the relational model, thereby necessitating a deeper analysis than an examination of the public interface part.

In some cases, an object class can map to a domain, and in others, to a relvar-like entity but not to a relation. If the object class represents basic, atomic values, it can map directly to a single domain. However, by virtue of the principle of encapsulation, nothing in the object-oriented model prevents an object class from being quite complex, in which case the object class could map to a composite data type, consisting of two or more atomic data types.

Although this composite type lacks some characteristics of a relvar, it can be treated as a relvar precursor in which the domains, but not the attributes have been defined. Consider the following definition of complex domain D, which is a composite data type consisting of a collection of single, atomic domains, V:

(3) $$D = \{ V_1, V_2 \ldots V_n \}$$

Note that D is analogous to relvar, R without attributes, A. In fact, the scalar data types that compose D can suggest attribute names without D actually being a relvar. The concept of a composite data type or complex domain plays a key role that leads to the generic relational data model which is the intermediate step in the proposed mapping scheme. Note that there is nothing to prevent "n" from being equal to 1, in which case D would degenerate into a scalar, atomic domain. D fits the description of the relvar-like entity mentioned above.

Although the domain of the relational world that corresponds to the object class of the object-oriented world, it is necessary to allow for the aggregation of data types to form complex domains. This is accomplished with the concept of data abstraction. Object classes that represent scalar data types map to atomic domains in the relational model, can be used in the construction of relations. Similarly, object classes that represent a composite data type map to complex domains. Relations can be constructed from these complex data types by choosing appropriate attribute name and data element values associated with each component domain. This is not the only way to form relations from complex domains because a relation also can take into account the information contained in message communications between

objects. Certain aspects of the message communication between objects in these classes may be expressed as relationships that also can be captured in relational format, if needed by the user. Messages can represent relationships between objects that may not be explicit in the object classes themselves, but that may prove useful if included in a relational design. Thus, message communications and also can be used to influence relvar structure. It is possible to express all aspects of the object-oriented model in terms of the relational model, either explicitly or implicitly.

APPENDIX A REFERENCES

[A1] H. Darwen and C. J. Date, "Introducing the third manifesto," *Database Programming and Design*, Vol. 8, No. 1, pp. 25 - 35, January 1995.

[A2] H. Darwen and C. J. Date, "The third manifesto," *SIGMOD Record* , Vol. 24, No. 1, March 1995.

[A3] C. J. Date, "Domains, relations and data types," *Database Programming and Design*, Vol. 7 No. 6, June 1994.

AUTHOR BIOGRAPHIES

Dr. Marion G. Ceruti

NCCOSC RDT&E Division, Code 4221
53140 Gatchell Rd., San Diego, CA 92152-7464
Tel. (619) 553-4068, DSN 553-4068, Fax (619) 553-5136
INTERNET: ceruti@nosc.mil

Dr. Ceruti is a scientist in the C^4I Systems Engineering and Integration Group of the Command and Intelligence Systems Division at the Naval Command, Control and Ocean Surveillance Center, Research, Development, Test, and Evaluation Division. She received her Ph.D. in physical chemistry, with emphasis on data acquisition systems, from the University of California at Los Angeles (UCLA) in 1979. While at UCLA, she was awarded a research fellowship from the International Business Machine Corp. Dr. Ceruti's present professional activities include database development and integration for C^4I decision-support systems, including the Joint Maritime Command Information System. She has served on the program committee and as Government Point of Contact for all annual Database Colloquia since 1987. An active member of AFCEA and several other scientific and professional organizations, Dr. Ceruti is the author of numerous publications on various topics in science and engineering, including information management.

Dr. Magdi N. Kamel

Department of Systems Management
Naval Postgraduate School
555 Dyer Rd., Monterey, CA 93943
Tel. (408) 656-2494, Fax (408) 656-3407
INTERNET: kamel@nps.navy.mil

Dr. Kamel is an Associate Professor in the Information Systems Group in the Naval Postgraduate School. He received his Ph.D. in information systems from the Wharton School, University of Pennsylvania. His main research interests include database management systems, specifically data models and languages, interoperability, and integration issues in heterogeneous databases, and the integration of databases with expert and decision support systems. Dr. Kamel is frequently invited to present papers on this subject at meetings and conferences. He has consulted in these areas for several organizations and is the author of numerous published research papers on database management topics. Dr. Kamel is a member of Association for Computing Machinery and the IEEE Computer Society.

Dr. Bhavani M. Thuraisingham

Network and Distributed Information Systems Center
The MITRE Corporation
202 Burlington Road, Bedford, MA 01730-1420
Tel. (617) 271-8873, Fax (617) 271-2352
INTERNET: thura@mitre.org

Dr. Thuraisingham is a principal engineer in the Network and Distributed Information Systems Center of the MITRE Corporation. She received the M. Sc. degree from the University of Bristol, U. K. and the Ph.D. degree from the University of Wales, Swansea, U. K. At the MITRE Corporation, she heads the Corporate Initiative on Data Management Research. Her research interests include heterogeneous database integration, database security, real-time database systems, massive multimedia database management, and object-oriented design and analysis techniques for developing various information systems applications.
Dr. Thuraisingham's work has been published in over 40 journal papers and in several conference proceedings. She has coedited one book on secure database systems and another on object-oriented systems. She delivered the featured technical address at the DOD Database Colloquium in 1994, and has served on the editorial boards of two computer journals. An inventor, she was awarded a patent on database inference control. Dr. Thuraisingham is a member of Association for Computing Machinery, the IEEE Computer Society, and the British Computer Society.

DATABASE INTEGRATION USING NWTDB PROCEDURES

by R. Gressang, G. Michaels, E. Harris, J. Mathwick, and J. Lu
SWL Division, GRC International, Inc., Vienna, VA

ABSTRACT

Naval and Joint Warfare in the future will require data on enemy, friendly, and neutral forces to be processed without error or delay. To achieve this goal, the Naval Warfare Tactical Database (NWTDB) Process is developing common data standards and structures that blend National Joint, and Naval information requirements and data standards. Central to the NWTDB Information Engineering Process is establishing a logical data model for producer databases by a process of extracting logical data models from existing physical database models. The NWTDB process not only results in a logical model for the integrated databases, it also provides traceability to the producer and operational physical databases which have been integrated.

The basic NWTDB Information Engineering Process consists of the following steps:

- Input existing database data elements, system data elements, and system structures into a common structure
- Convert to a common frame of reference which is recorded in the Systems Information Directory (SID)
- Compare and analyze structures and definitions
- Output structures, a Data Element Dictionary, and models back to originators for review and comment on interpretation
- Make corrections based on originator's review
- Conduct desktop publishing of DED and structures for standards manual
- Transfer the data from the NWTDB data element template into a standard data element template and submit to DDRS.

To support the NWTDB process, a methodology for integrating existing databases has been developed in which the legacy physical databases are converted to a common frame of reference and the structures and definitions compared and analyzed. The conceptual basis used for integration is based upon applying subject matter expertise in a systematic approach drawing upon classical systems engineering.

The entire NWTDB process, to be practical, must be supported by a full set of automated tools. In the evolution of the NWTDB process, no single set of Information Engineering case tools has been found which is able to support the entire process. As a result, a client/server environment providing an array of commercially available CASE tools integrated with specially developed applications has been developed. As no currently available tool provided these capabilities, a PC based tool called Data Analysis and Reconciliation Tool (DART) was developed to support recording of legacy system physical data element definitions, mapping those physical data element definitions to a logical model, and developing normalized data elements in accordance with DoD Directive 8320.1.

1. INTRODUCTION

Naval and Joint Warfare in the future will be conducted in a dynamic, information intensive warfare environment. Data on enemy, friendly, and neutral forces will need to be processed without error or delay. Due to the rapid evolution of information technologies, the systems that will process the required information need to be designed for evolutionary insertion of new technology. Interoperability of the information systems supporting the operational forces will be essential.

To achieve this goal, the Naval Warfare Tactical Database (NWTDB) Process is developing common data standards and structures that blend National, Joint, and Naval information requirements and data standards. The NWTDB Process is implementing guidance provided by DoD Directive 8320.1, DoD Data Administration, and will result in the data standards and structures used in Maritime operations being recorded, registered, and connected via the DoD C2 Core Data Model to the DoD Enterprise Data Model. .

2. NWTDB PROCESS

Central to the NWTDB Information Engineering Process is establishing a logical data model for producer databases by a process of extracting logical data models from existing physical database models. These extracted logical models, through application of operational knowledge, operational information engineering, and normalization are continuously converged into a single, ever expanding, logical model. The NWTDB logical model is related to DoD Enterprise Models by establishing it as an extension of the DoD C2 Data Model. Data. Tactical decision aids and other operational requirements and systems are represented as views of the data in the NWTDB logical model.

The NWTDB process not only results in a logical model for the integrated databases, it also provides traceability to the producer and operational physical databases which have been integrated. This traceability allows determining, for each standardized data element in the logical data model, whether: (1) instance fill exists and is used (2) instance fill is produced but is not used (3) instance fill is not produced but would be used if it were produced and (4) instance fill is not produced and would not be used if it were.

The basic NWTDB Information Engineering Process consists of the following steps:

- Input existing database data elements, system data elements, and system structures into a common structure
- Convert to a common frame of reference which is recorded in the Systems Information Directory (SID)
- Compare and analyze structures and definitions
- Output structures, a Data Element Dictionary, and models back to originators for review and comment on interpretation
- Make corrections based on originator's review

- Conduct desktop publishing of DED and structures for standards manual
- Transfer the data from the NWTDB data element template into a standard data element template and submit to DDRS.

3. NWTDB OPERATIONAL INFORMATION ENGINEERING

The process for integrating existing databases occurs in the steps where the databases are converted to a common frame of reference and the structures and definitions compared and analyzed. The conceptual basis used for integration is based upon applying subject matter expertise in a systematic approach drawing upon classical systems engineering. The steps involved are described in the following paragraphs.

Step One

The first step is to examine the legacy databases to be integrated to determine 'what' and 'why' from the perspective of a user of the database. The focus here is on developing answers to two basic questions. The first of these questions is 'what is the nature of the object or process which is being described by these attributes?" The second question, equally important, is to determine the reason that these particular attributes are being used to describe the object or process, i.e. what is the view of the object or process held by the person recording these attributes. As an example to be carried through this part of the paper, consider the weapons systems data structures relating to missiles and ships in the Naval Intelligence Database (NID), one of the databases in the NWTDB Standards Manual. The attributes recorded about a missile support defining a missile as an unmanned weapon which is steered to its target after it is launched, while the attributes recorded about a ship establish that its important characteristics are that it is a vessel for traveling on the surface of or under the surface of a body of water. The reason these attributes are being recorded is to provide reference data and background information for planning Naval operations. Considering the attributes recorded for ships, the recorded information is very specific related to weapons associated with warships and the cargo capacity of merchant ships, information of great value for planning naval operations. However, it is not of sufficient detail to be used for engineering calculations. For example, while many attributes are recorded on ship dimensions, ship engineering, and ship propulsion, additional data would be required to perform ship powering calculations.

Step Two

Based upon the results of the first step and the pertinent technical or operational area characteristics, the second step is to perform a functional analysis (in the systems engineering sense of the term) and synthesize a 'generalized system' to a first level of functional allocation(also in the systems specification sense). Systems Engineering here refers to the processes described by the following extract from MIL-STD-499A, "the application of scientific and engineering efforts to (a) transform an operational need into a description of system performance parameters and a system configuration through use of an iterative process of definition, synthesis, analysis, design, test, and evaluation; (b) integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system

definition and design;." The functional analysis is used to identify a baseline of functions and a list of representative functional performance requirements which would be expected to be associated with those functions. The system synthesis seeks to determine the configuration and arrangement of a system which satisfies the functional performance requirements. In the operational information engineering context, these analyses do not have to be elaborate or complex. The only functional requirements which have to be considered are those relevant to the view of the users of the database, and the system synthesis proceeds only to the level of identifying what would be the major 'configuration items' of a system. The attributes recorded in the legacy databases which were analyzed in the first step will be used to define the detail of any further 'generalized system' structure.

Continuing the examples involving ship and guided missile, the end results are 'generalized systems' as follows:

Ship 'generalized system" configuration items:

- Identification, Function, & History
- Physical Description, Capacity, & Personnel
- Engineering, Propulsion, & Electrical
- Electronics, Antennas, & Sensors
- Embarked Vehicles
- Weapons & Countermeasures
- Performance & Operational Factors

Guided Missile 'generalized system' configuration items:

- Identification
- Physical Description
- Performance Envelope
- Operational Characteristics
- Warhead
- Guidance
- Propulsion
- Signatures
- System Integration

Step Three

The third step is to allocate and align legacy database entities and attributes to the 'generalized system' to test for a conceptual match, adjusting the 'generalized system' as necessary to achieve a match. In a sense, the skeleton outline of a specification for the 'generalized system' is filled in, using the metadata for database entities and attributes as the source of the details. Also at this stage, if it has not already been decided, a decision should be made as to where and how to attach the entities evolving from the operational information engineering process to the overall enterprise data model.

For example, the Naval Intelligence Database (NID) contains five structures (containing information on ship classes, individual ships, merchant ships, submarine classes, and

individual submarines) with a total of 178 entities which can be mapped to the ship 'generalized system.' Focusing on the Engineering ,Propulsion, & Electrical aspects of the ship 'generalized system', the result of allocating and aligning NID Ship Entities to the 'generalized system' Engineering, Propulsion, & Electrical configuration item aligns with Engineering & Propulsion the following entities: SHIP-REACTORS, SHIP-ENGINEERING,SHIP-ENGINES,SHIP-BOILERS,ASHIP-REACTORS,ASHIP-ENGINEERING,ASHIP-ENGINES, ASHIP-BOILERS, MER-SHIP-ENGINES, SUB-REACTORS, SUB-ENGINEERING, SUB-ENGINES, SUB-SHAFTS, ASUB-REACTORS,ASUB-ENGINEERING,ASUB-ENGINES,ASUB-SHAFTS, and with Electric the following entities: SHIP-ELECTRIC, SHIP-GENERATORS, SHIP-ENGINEERING, ASHIP-ELECTRIC, ASHIP-GENERATORS, ASHIP-ENGINEERING, MER-SHIP-GENERATORS, SUB-BATTERY, SUB-ELECTRIC, SUB-GENERATORS, SUB-ENGINEERING, ASUB-BATTERY, ASUB-ELECTRIC, ASUB-GENERATORS, ASUB-ENGINEERING. Connection of both the new ship and guided missile entities to the DoD Enterprise Model as subtypes of MATERIAL-TYPE in the C2 Core Data Model best fits NID user's view of these entities.

Step Four

The fourth step is then to determine legacy database entities containing common attributes. This is a process of carefully examining the metadata and definitions associated with the attributes, to determine and understand the underlying concepts for each attribute. The physical concepts found in the definitions of the attributes, dimensional analysis to uncover differences in physical concepts, applying operational experience as to what data is immediately useful operationally and what data is required by tactical decision aids, data requirements for modeling and simulation algorithms, and engineering design models and data (such as missile guidance system block diagrams, IR detector material tabulated properties, calculation procedures for determining ship displacement and capacity, and ship powering calculations) all provide insight into whether attributes are common. The context for the metadata provided by the examination is used in conjunction with how a database user would use the attributes to decide if they are common. As an example of this, when attributes related to ship electrical systems found in the NID entities listed above are examined, the two attributes of PWR-SUPPLY-TOTAL-KW and PWR-GENERATOR-TOT-KW are found to both represent the maximum electrical power not used for propulsion available on a ship.

Step Five

The fifth step is to allocate and align legacy database attributes to new entities based on the 'generalized system' configuration items. In this process, the realigned attributes are examined to see if natural conceptual groupings exist, which can form the basis for defining new entities. At this point also, issues such as the cardinality of the relations between the new entities should be identified and resolved. The output of this step is a listing of proposed new entities, and for each proposed new entity a listing of its proposed attributes and its relationships with other entities, including cardinality. As an example, attributes spread among the NID tables called SHIP-ELECTRIC, SHIP-

GENERATORS, SHIP-ENGINEERING,ASHIP-ELECTRIC, ASHIP-GENERATORS, ASHIP-ENGINEERING, MER-SHIP-ELECTRIC, MER-SHIP-GENERATORS, SUB-ELECTRIC, SUB-GENERATORS, SUB-ENGINEERING, SUB-BATTERY, ASUB-ELECTRIC, ASUB-GENERATORS, ASUB-ENGINEERING, and ASUB-BATTERY can be grouped into two proposed entities, SHIP-ELECTRIC (which can be a child of a SHIP-ENGINEERING entity) and SHIP-BATTERY, a child of the SHIP-ELECTRIC entity. SHIP- ELECTRIC would have the attributes TYPE-CURRENT, VOLTAGE-V, FREQ-CURRENT-HZ, PHASE-CURRENT, GENERAL-RMKS, TYPE-GENERATOR,DESIG-GENERATOR, PWR-GENERATOR-KW,NBR-ABOARD, and PWR-SUPPLY-TOT-KW, while SHIP-BATTERY would have the attributes TYPE-BTRY, DESIG-BTRY, NBR-BTRY-GROUPS, NBR-BTRY-CELLS-PER-GROUP, RATE-DISCHG-LO, RATE-DISCHG-HI, and GENERAL-RMKS. Associated cardinalities would be one to one between SHIP-ENGINEERING and SHIP-ELECTRIC, and one to zero or one between SHIP-ELECTRIC and SHIP-BATTERY.

Step Six

The sixth and final step in the process is to check to ensure every legacy database attribute is mapped into the new 'generalized system' entity structure and is represented by one of the proposed new attributes. This step ensures that all information being stored in the legacy databases can continue to be stored in the proposed new structures. This step is also useful as it provides a check on the completeness of the restructuring, ensuring that no concepts are being overlooked, and it also provides a mechanism for user's of the legacy databases to relate the structures that they are familiar with to the proposed new structures.

The result of this process is a new entity/attribute structure incorporating and based upon subject matter expertise, which serves as an input to further technical data modeling activity. This new structure is ready for the rigorous application of normalization rules and algorithms, with the resulting entities, associated attributes, relationships and cardinality providing a formal IDEF1X model of the new structure. The resulting IDEF1X data model can also provide the initial inputs required to develop DoD Data Standardization Package Proposals.

4. AUTOMATED TOOL SUPPORT

The entire NWTDB process, to be practical, must be supported by a full set of automated tools. In the evolution of the NWTDB process, no single set of Information Engineering case tools has been found which is able to support the entire process. As a result, a client/server environment providing an array of commercially available CASE tools integrated with specially developed applications has been developed. This client/server environment, developed using an open architecture philosophy, is called the Systems Information Directory (SID). SID's capabilities are to:

- Collect, store, analyze and manage the metadata life-cycle for non-homogeneous information systems and databases
- Reverse engineer data on existing physical systems and normalize data

- Maintain a logical data model for databases and map physical system data elements to it
- Perform impact analysis and produce analytical reports
- Provide full data traceability throughout an information system's life-cycle
- Support data element registration in the DDRS

SID has been developed under a philosophy of not developing a tool if an existing available tool would suffice. Thus it has incorporated various commercially available tools for as many functions as possible. For example, IE Expert/Advantage is used to support business rule development, forward engineering, and reverse engineering. ERWIN/ERX is used for IDEF1X data modeling, and BPWIN provides an IDEF0 process modeling capability. Data Flow diagramming, node trees, and complex data are supported by ABC Flow Charter. Object View is used for Window/GUI development, SQL linkage to databases, and for 4GL language. Power Viewer is also used for SQL linkage to databases and for report generation, augmenting Object View.

Despite this extensive incorporation of existing tools, however, not all required functions can be performed using commercially available tools. Examples of functions which required the development of a special tool are producing structures, data element dictionaries, and models in DDRS formats for review and comment by the owners of the legacy databases, and recording in a systematic manner the linkages between legacy system data elements and the attributes of the NWTDB data model. The volume of metadata to be produced and reviewed requires use of automated tools for efficiency, while accurately recording the linkages between logical data model attributes and physical database data elements is essential to provide the traceability discussed earlier. As no currently available tool could be found which provided these capabilities, a PC based tool called Data Analysis and Reconciliation Tool (DART) was developed for this purpose. DART has just been released and is undergoing beta testing. DART is a Microsoft Windows™ based application supporting the recording of legacy system physical data element definitions, mapping those physical data element definitions to a logical model, and developing normalized data elements in accordance with DoD Directive 8320.1.

Biographical Sketches of the Authors

Dr. Randall Gressang is currently Chief Scientist of SWL, Inc. His telephone number is 703-506-5769, FAX number is 703-506-0585, and e-mail address is Randy_Gressang@va.grci.com. He is involved in a wide variety of database, signal processing, telecommunications, and instrumentation projects. He has over 27 years experience in the areas of C3I, Radar Systems, Space Systems, High Performance Computing, and Radio Navigation.. He is a retired Air Force Officer, and prior to joining SWL was Technical Assistant to the Director of ARPA and involved in C3I and Distributed Simulation. Before being assigned to ARPA, he served in a variety of positions in the USAF. He received his Ph.D. from AFIT, an M.S. from MIT, and a BSE from Princeton, all in Engineering.

Mr Gregory Michaels is Director of the Information Management Division at SWL. His telephone number is 703-506-5811 and FAX number is 703-506-0585. He has over 30 years experience in the area of Navy

Data Interoperability Between C^{3}I Systems

Scott A. Renner
Arnon S. Rosenthal
James G. Scarano

The MITRE Corporation

1. DATA INTEROPERABILITY: THE PROBLEM

The *C4I for the Warrior* concept includes the "infosphere," which is the union of all information sources, fusion centers, and distribution systems. In the future, the infosphere will supply information to users in the form of a pre-planned "push", and also as a response to run-time queries as users "pull" information for new and unforeseen needs [JCS92]. In the present, C^{3}I systems merely send information in the form of standard messages to a few, pre-arranged partners. Both contemporary systems and those of the future must deal with the same issue: one system has information, and others need to have it. The problem we address is that the same *information* may not be the same *data* on different systems, as illustrated in the following figure.

System A

Table: ACMAINT

ACTYPE	RDYWHEN	NUM
F15	0500	22
F16	1700	16

System B

Table: RDYACFT

MODEL	AVAILTIME	QTY
F15	0500	22
F16	1700	16

System C

Table: MAINTSCHED

RDYTIME	F15S	F16S
0500	22	-
1700	-	16

System D

Table: RDYF15S

WHEN	QUANTITY
0500	22

Table: RDYF16S

WHEN	QUANTITY
1700	16

System E

Table: ACMAINT

ACTYPE	RDYWHEN	NUM
F15	5:00A	22
F16	5:00P	16

Figure 1: Same information, five separate data schemas

Figure 1 shows five different data schemas, all representing the same information about aircraft maintenance schedules. Systems A and B are different only in the names of the data elements. Systems B, C, and D have a different structure: the type of aircraft is represented as a value in system B, an attribute name in system C, and a table name in system D. Finally, systems A and E are structurally equivalent, but have a different representation for the scheduled completion time. In order for any of these systems to exchange information, the data must be converted from the source schema to the receiver schema.

(A related problem, and one which we do not attempt to solve, is that the same *data* may not be the same *information* in the source and receiving system. For example, it does no good to have a complete match in name, structure, and representation of data, if one system is scheduling aircraft for maintenance and the other is scheduling aircraft for combat missions. All we can do with this kind of semantic mismatch is try to detect and avoid it.)

The current approach to the data interoperability problem is to write *ad-hoc* data interface programs for each pair of communicating systems. Experience shows that development and maintenance of these programs is expensive in terms of both time and money. Worse, the total effort required increases with the *square* of the number of communicating systems. Finally, these hard-coded interfaces support only the information transfer anticipated during development, and not the "pull-on-demand" transfers anticipated in the infosphere. It is plainly evident that the current approach cannot be made to support the requirements of the infosphere, or even those of the immediate future.

2. DATA STANDARDIZATION IS NOT THE (WHOLE) SOLUTION

If every system always used the same data to represent the same information – identical names, structure, and representations – then the data interoperability problem would go away. The DOD data standardization program will do this to some extent, but there are reasons why standardization will not be a complete solution, which we will consider in this section.

Certain kinds of metadata resist standardization.

A key notion in data element standardization is that there is exactly one data element for each "fact" – or more specifically, for each attribute of each entity in the data model. Also, the data element metadata must be fully specified, so that all of the properties of the data element are known and standardized [DOD93]. This facilitates information exchange between systems that use a standard data element. No translations are required, because the name and representations are known to be the same.

We believe that there are legitimate reasons why systems might need different metadata for the same attribute. Imagine two systems which keep track of the time at which some event is observed. System A makes its observations with a very precise electronic sensor, while system B receives event times from a variety of human observers. System B can use the observations of system A, but not vice versa. Within the context of the data model, the two systems are recording information about the same fact, so we would like them to use the same data element. However, the metadata can't be the same: system A data has a higher precision and very likely a higher confidence than system B data.

There are three possible solutions to this problem. First, we can ignore it, by causing both systems to use a data element defined at the higher precision. System B simply ignores the extra precision digits in its database, probably filling them in with meaningless zeroes. This amounts to

telling a lie about the data. The trouble is that sooner or later, the lie is going to be believed. Some system will receive system B data and treat it according to the high precision that it claims, not the low precision that it actually has.

A second solution is to introduce new data elements to explicitly represent the different metadata. That is, in addition to the time-of-event data element, the data model could have a precision-of-time-of-event and a confidence-in-time-of-event data element. This has a good chance of avoiding the probable data-accuracy error in the first solution. However, it causes the two databases to be filled with data that neither system actually needs. (For example, in system A, the precision and confidence data elements *always* have the value "high.")

The third solution is to create separate, fully-defined data elements for each combination of metadata. Systems A and B may then have distinct, properly-defined data elements for recording their time-of-event data. We avoid the data-accuracy error and avoid introducing unnecessary data elements. However, this approach may greatly increase the number of data elements in the data standard. If there are five possible precisions and five levels of confidence, then we need 25 "standard" data elements just for the time-of-event concept. Also, we introduce data interoperability problems. For information to flow from A to B, we need a human to notice the similarity in data elements and to supply the appropriate translation between them.

The third solution is the best of the three, but it results in an overly-complex standard data model and puts too much of the interoperability burden on the system developers. We believe that a better approach is to relax the fully-specified-metadata requirement, permit system developers to specify the metadata used in their data schema, and provide tools for automatically resolving metadata differences. We will return to this theme in section 3.

Constructing and maintaining a single, integrated standard data model is difficult or impossible.

For every system to use the same data to represent the same information, we need a single, integrated data model covering the union of the system domains. This approach can work for small, simple enterprises. However, the DOD is neither small nor simple. Purely from the standpoint of human limits on comprehension, we should not expect success in constructing a single model of appropriate detail for a large enterprise. Instead, we should expect many models, each covering a single functional domain. Systems will adopt data definitions from the appropriate model. This introduces data interoperability problems wherever systems communicate across the boundaries of separate models.

The standard will change, but systems will not all simultaneously change to conform.

Even if we somehow obtained a single data model, system developers would immediately press for incompatible modifications and extensions. This is unavoidable: as the world changes, so must our representations of it. In large-scale models, these changes will be frequent. For example, comparatively stable databases might require one schema change every three years. A standard model covering 100 such databases must cope with a change every two weeks.

Every change to the standard data model will require changes to all of the affected systems. These modifications can require a great deal of time and money to perform. It is highly unlikely that all of these systems can coordinate their changes to take effect simultaneously. Instead, systems will come into compliance over time. This introduces data interoperability problems between the systems that have changed and those that have not.

There will always be a requirement to communicate with non-conforming systems.

No data standard is of any use when your communication partners have not adopted it. We believe that C^3I systems will always be required to exchange information with systems that do not conform to the DOD data standard. We do not expect that allied systems (e.g. the French army) and commercial systems (e.g. United Airlines, Wal-Mart) will be adopting the DOD data standard any time soon. We also believe that some DOD legacy systems will be around much longer than many people expect. Information exchange with these systems introduces all of the data interoperability problems mentioned above.

3. THE DATA MEDIATION APPROACH

We are developing a *data mediation* approach to solve the data interoperability problem. A data mediator is a computer program which translates data between two systems with different data schemas. In our approach, the mediator handles an information exchange between a source and receiver system in two steps. Beginning with a query from the receiver's schema, we first translate it into the equivalent query against the source schema. Then, we execute the source query and translate the retrieved source data into the receiver's format. The result is that the mediator acts as a *semantic gateway* between the systems, permitting the receiver to view the source as an extension of its own database, without concern for the differences in names and representations of data.

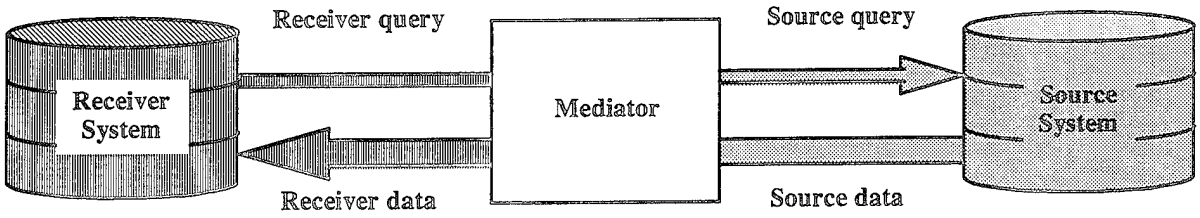


Figure 2: Data mediation

The difference between a mediator and the hard-coded translators in use today is that the mediator automatically generates data translations from descriptions of the data in the source and receiver schemas. These descriptions must have the following properties:

- They must be written using a *formal language*. Natural language descriptions may be sufficient for human developers, but not for the mediator program.

- They must be written using a *common vocabulary*. System developers must agree on the meanings of the terms in their descriptions in order for the mediator to identify and correlate related elements in different schemas.

- They must be *adequate* for the mediation that is required. (It is not necessary that they be *complete*. Those aspects of a data schema which are never exchanged with another system need not be described at all.)

Our approach depends on a shared, conceptual reference schema (or ontology), as in the Carnot project [Collet91] and the SIMS project [Arens92]. The descriptions of the source and receiver schemas (collectively, the *component schemas*) use the terms defined in the reference schema as their common vocabulary. In our implementation, the reference schema is composed of an IDEF1X data model covering the functional domain plus a library of data element conversion functions. The component schema descriptions are expressed as database views which show the correlations between component and reference entities, plus enough semantic information about each component data element to permit the mediator to select the proper conversion functions.

Figure 3 shows two component schemas and their correlations to a common reference schema. The figure shows the descriptions necessary for an information flow from system SRC to system RCV. (A bi-directional flow is possible, probably will be routine, and requires similar descriptions which are omitted from this example.) The correlations between the source schema and the reference schema are expressed in terms of a *source view* which is defined in the source database. The source view explains how to retrieve the data for a reference schema entity from the source database. Correlations between the receiver schema and the reference schema are expressed in terms of a *receiver view*, which explains how to retrieve the data for a receiver relation from the reference schema.

In addition to the schema correlations, each attribute in the component schemas is annotated with the metadata describing its meaning and properties (e.g. precision, units of measure, quality, etc.) These properties are the *meta-attributes* for the data element, and collectively form the definition of its *semantic domain*. The list of meta-attributes is defined in the reference schema; this supplies a common vocabulary for describing the meaning of data elements. When corresponding data elements in the source and receiver schemas have different semantic domains, the mediator searches its library of conversion functions to compose a sequence of calls which eliminate the differences, as in [Sciore94].

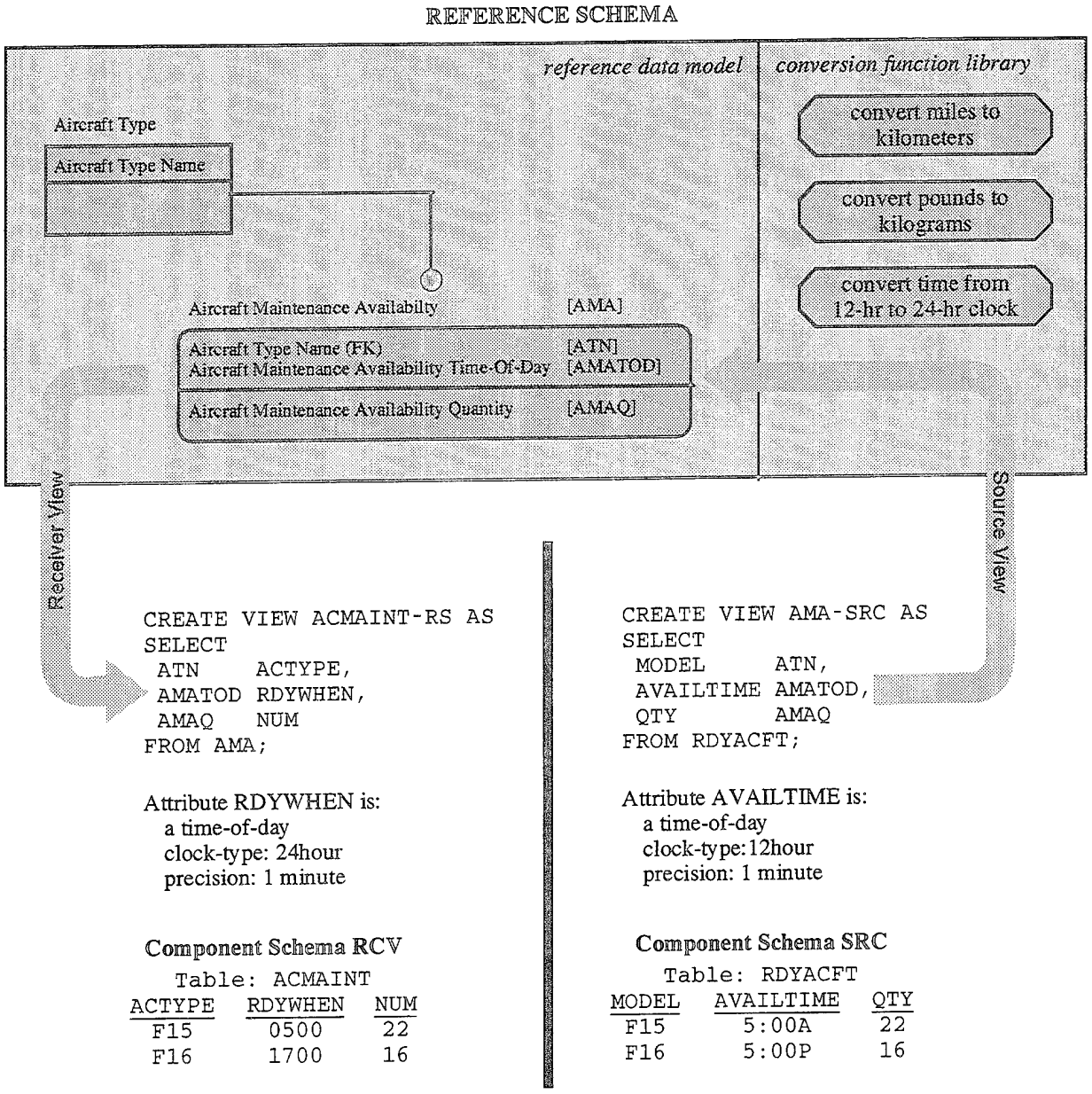


Figure 3: Reference schema and component schemas

The following (simplified) example shows how the mediator operates. The mediator is given an input query, which is written in terms of the receiver's schema, and is directed to retrieve the corresponding data from the source database. In our example, the query is:

```
SELECT * FROM ACMAINT WHERE ACTYPE = 'F15';
```

In the first step, the mediator resolves the structural differences between the source and receiver schemas. The input query requests data from the ACTYPE table; the receiver view shows how to produce the ACTYPE table from the AMA entity in the reference schema; the source view shows

how to produce the AMA table from the RDYACFT table in the source schema. By composing the two views, the mediator forms a single query which can be executed on the source database to produce a new table, based on the receiver's relations, but populated with source data values. This step resolves differences in data names and structure, but not representation. The created table looks like this:

ACTYPE	RDYWHEN	NUM
F15	5:00A	22
F16	5:00P	16

In the second step, the mediator translates the retrieved values into the context expected by the receiver. The mediator iterates through the retrieved tuples, comparing the source and receiver contexts of each value, and applying an appropriate conversion function from its library to compute the receiver's value. In this case, only the RDYWHEN attribute requires translation, and the mediator applies the function which converts time from a 12-hour to a 24-hour clock. (In our example, the conversion function for the RDYWHEN attribute can be determined in advance by inspection of the query; in the general case, the mediator may need to modify the composed query in order to carry the source's context information along with its data values.) This second step creates a new table containing source information in the receiver's context. In our example, the resulting table looks like this:

ACTYPE	RDYWHEN	NUM
F15	0500	22
F16	1700	16

The final step is to apply the input query to the generated table and return the results. In our example, the final output is :

ACTYPE	RDYWHEN	NUM
F15	0500	22

An important aspect of our approach is that, while the reference schema is shared, the labor of producing component schemas is not. System developers produce their schema descriptions independently. Only the mediator has to see both descriptions at once.

The mediator can be used either by system developers or by end users. For developers, the mediator acts as a tool for creating data mediation software for information exchange with a partner that is known in advance. Developers obtain the schema description of the other system, add the description of their own system, and specify the information to be sent and/or received. The mediator then generates a data interface program which is compiled, distributed with the rest of the system, and executed whenever the systems must communicate. For users, the mediator operates at run-time to handle arbitrary queries against another system's database. The mediator translates the user's query and the retrieved data, displaying the result to the user in the format of his host system.

4. ADVANTAGES OF DATA MEDIATION

We believe that our approach is complementary to the DOD data standardization program. Data standardization will reduce the need for mediation, but will not eliminate it. The standard data models developed as part of the 8320.1 process can serve as the backbone of the reference schemas required in our mediation approach. We expect that in time we may identify some additional metadata that should be collected for standard data elements.

The component schema descriptions produced for our approach are *reusable*. Developers describe their data schema once; the description can then be used by the mediator to communicate with as many other systems as required. This solves the interface explosion problem. It does not matter how many data interface programs are generated by the mediator; the work done by developers — writing descriptions of their data schemas— grows linearly with the number of communicating systems.

The component schema descriptions are easier to write and maintain than the equivalent interface code. It is less work to write n descriptions than to hand-code n^2 interfaces. We also believe that it is simpler, and more efficient to write and maintain data knowledge in a declarative description, than to maintain the same knowledge embedded in the procedural source code of a data interface.

The component schema descriptions can serve as precise, reliable documentation for system developers and users. The traditional system of free-text comments is notorious for outdated, imprecise documentation. Because our descriptions are used to produce system software, they must be kept current. Because our descriptions are the input to a computer program, their meaning must be precisely defined.

Data mediation allows individual systems to keep their own "view of the world." Users are typically reluctant to abandon their own data schema in favor of a standard schema supplied by someone else. The infosphere is supposed to supply information to users in the form they require; data mediation allows users to specify "what they need and how they need it." Our approach allows users to keep their schema so long as they can describe it to the interface generator. Our approach is 100% "carrot" — use the mediator, because it makes your information exchange tasks simpler. Data standardization is 100% "stick" — adopt the data standard, or (eventually) lose your program funding.

Unlike many other proposals for integrating heterogeneous databases, our approach works as a layer on top of the existing DBMS systems. It does not require a new query language, or the installation of new, leading-edge DBMS software. We are concentrating now on relational database systems. However, we believe that our approach can be extended to work for other database models, as well as for generating standard message format interfaces.

5. SUMMARY

Data interoperability is a problem for C^3I systems now, and will remain a problem in the future. Data standardization helps, but is not the whole solution. We are developing a data mediation approach which complements the data standardization process, supplies the missing parts of the solution, and can be integrated with existing C^3I database systems. We believe that our data mediation approach will provide a flexible, cost-effective mechanism for satisfying the information exchange requirements of all types of systems. At present, the capabilities of our data mediator are limited, and there are issues yet to be resolved. We will demonstrate the feasibility of our approach during the Joint Warrior Interoperability Demonstration (JWID) in September, 1995.

6. REFERENCES

[Arens91]
Arens, Y., and Knoblock, C. A. Planning and reformulating queries for semantically-modeled multidatabase systems. In *Proceedings of the 1st International Conference on Information and Knowledge Management* (1992), pp. 92-101.

[Collet91]
Collet, C., Huhns, M. N., and Shen, W. M. Resource integration using a large knowledge base in Carnot. *IEEE Computer,* Vol. 24, No. 12, December 1991.

[DOD93]
Department of Defense, *Data Element Standardization Procedures*, January 1993. DOD 8320.1-M-1.

[JCS92]
Joint Chiefs of Staff, *C4I for the Warrior Objective Concept,* September 1992. Coordination draft.

[Sciore94]
Sciore, E., Siegel, M., Rosenthal, A. Using semantic values to facility interoperability among heterogeneous information systems. *ACM Transactions on Database Systems,* June 1994.

Biographical Notes

Scott Renner is a Lead Engineer in the Combat C^3 Systems department at MITRE. His research interests include object-oriented programming, machine learning, and automated program debugging. He received a Ph.D. in computer science from the University of Illinois at Urbana-Champaign in 1990. Contact information: sar@mitre.org, (804) 766-4592.

Arnie Rosenthal is a Lead Scientist in MITRE's Center for Integrated Intelligence Systems. He has published numerous papers in the areas of database design tools and theory, active databases, query processing, and computational complexity. He has a Ph.D. from the University of California at Berkeley. Contact: arnie@mitre.org, (617) 271-7757

Jay Scarano is a Group Leader in the Development and Methodology/Tools department at MITRE. He has a BSEE from the University of Hartford, College of Engineering. His research interests include heterogeneous databases, distributed object management, and fault-tolerant distributed systems. Contact: jgs@mitre.org, (617) 271-3979.

A Model for Information Retrieval from Heterogeneous Sources

A. Ruocco, O. Frieder

ABSTRACT

The old adage that "knowledge is power" has never been more true than in today's environment. The Gulf War clearly demonstrated the advantages of superior data gathering capabilities. The National Defense University has developed an entire curriculum around information based warfare. Data collection has experienced a virtual explosion. Requirements measured in gigabytes and terabytes are real planning figures. Data access must account for diverse databases, from traditional databases to message handling systems to action officer email. Even with massive storage systems, duplication of data from such a variety of data sources to one central location for "easier"access is no longer feasible. Increased communication capabilities and acceptance of tools such as Mosaic and the World-Wide Web make it clear that information is stored and must be retrieved from independent locations. An information model is needed which goes beyond the storage and distribution of data. The model must support the needs of senior decision makers and commanders in accessing data from sources they may not be familiar with. The model cannot be built under an assumption of replacing current systems. Instead the model must account for the independence of existing systems and data. It must strive for integration of data across sources, yet protect the needs of data owners to limit access to sensitive data.

We describe an information model which fulfills the requirements stated above. The model recognizes the need and desires of data owners to protect their data. At the same time, it recognizes the needs and desires of decision makers and commanders who have valid requirements to query outside sources. The model segregates a node into regions. A shared region allows interaction between an outside user and information the data owner is willing to share. The owned region belongs to the data owners. Information can be made available for sharing without effecting the owners underlying system. The model identifies a series of actions which take place within the shared region. In particular it discusses the ability for an outside query to enter the region, be transformed into a form that can be compared to information the owners provide. The results of the query can then be transformed into a form the owned system can respond to for subsequent document retrieval. The paper discusses some current research which justifies each step within the shared region.

The result is a model which does not impose a monolithic structure. It is built around current systems rather than a replacement of current systems. The model provides great flexibility for commanders and senior decision makers to find and access data from unfamiliar sources. It protects the needs and desires of data owners. Overall it allows the continued independence of distributed systems and supports the commander and senior decision maker's vital need for integration of data sources.

"A Model for Information Retrieval from Heterogeneous Sources"

Anthony S. Ruocco, Major, U.S. Army
Ophir Frieder, Dept. of Computer Science, George Mason University

1. INTRODUCTION

The Gulf War clearly demonstrated the advantages of superior data gathering capabilities. Satellites and other data collection technology, combined with data transmission technology make data collection measured in gigabytes and terabytes real planning figures. With current telecommunication capabilities, the ability to gather data at the foxhole level and pass it back to collection points via email is a reality. Various applications use these data to process them into a form that is meaningful to commanders and senior decision makers. For many applications and systems, "data-sharing" is merely the duplication of data from one source to another. Obviously, the growing magnitude of data makes simple duplication no longer viable. The commander's requirement is for every system supporting the operation to be able to integrate seamlessly with each other.

Technology and desires have out paced implementation of systems to meet demands. Data systems designed for independent operation are likely sources of information for a growing population. This population is growing in both numbers and diversity. The changing nature of operations shows the need to integrate information from military agencies, government agencies, and even non-government organizations for the senior decision makers to have a complete picture of the operation.

We describe an information model for information retrieval among heterogenous databases. The model provides the senior decision maker access to these information sources. The model recognizes the need for data owners to protect their data. It also recognizes that the vast amounts of existing data bases cannot be adjusted to fit into one single mold. Following a brief overview of information retrieval, the paper describes the structure of the model and some of the model processes. Justification of the model is based upon commonly accepted ongoing research efforts.

2. OVERVIEW OF INFORMATION RETRIEVAL

In its simplest form, information retrieval is the process of taking a query, examining a document set, and providing documents which satisfy the query to the user. The retrieval process uses the query as a basis of comparison and returns documents which approximate the query. How the underlying system approximates the query is based on one of four information retrieval models: text scanning, Boolean operations, document signatures, and a vector space model. Regardless of the underlying model, *recall* and *precision* are two measures of a query's success. *Recall* is the percentage of relevant retrieved documents to the total number of relevant documents in the database. *Precision* is the percentage of retrieved documents which are relevant to the total number of retrieved documents. Typically, the higher the *recall* the lower the *precision* and the higher the *precision* the lower the *recall* [15]. *Recall* and *precision* are

frequently based on tests against known databases. When faced with an unknown database, the user is unaware if all the relevant documents have been retrieved. That being the case, retrieval methods are frequently oriented towards getting the highest number of documents, i.e., high *recall* [7,9,11,13].

One of the principle methods of enhancing recall is to make use of a thesaurus. The system thesaurus is used to account for synonyms or to account for specific word or phrase usage. A thesaurus serves as a storage location for terms and their associated synonyms. The premise of a thesaurus can be expanded to include not just grammatical synonyms but related terms derived by usage such as parallel processing being related to concurrent processing. System thesauri are based on the document set a system maintains. Therefore, it is critical to account for an outside user coming in with a query that is based on some other, independent thesaurus.

An obvious solution to heterogeneous databases should be the merging of their thesauri. Unfortunately, that is not a simple task. Some words which are of little value to some have important meaning to others [1,4]. Studies have shown the probability of 2 persons using the same term to describe the same thing is less than 20%. The probability of two indexers assigning the same descriptors to the same document is between 10-20% [5]. Should each word and its grammatical or related terms be included? For example, the system would need to account for the usage of the word 'duck' as a small water fowl and as a quick, evasive maneuver. Where would such a monolithic entity reside and how would it be maintained? Such a global, all encompassing entity is just not feasible. Given the multiple usage of many words, it is clear that many times words only have significant meaning in how they are related to each other in the context of a particular document or query. In other words, it is the concepts the words describe rather than the words themselves which are important. In a heterogenous world a user would not know the source, therefore it is important to be able to separate the concepts from the data. One of the earliest successes in using independent concepts was the RUBRIC system.

RUBRIC (Rule-Based Retrieval of Information by Computer) was designed to capture the concepts of users in the users' own terms [13]. The authors describe the system as meeting many needs. First, queries can be expressed in natural language. Concepts can be readily modified by the user to handle changing interests. And, the use of concepts allows for partial matching of queries to documents. RUBRIC allows the user to define concepts then use those concepts to build a concept hierarchy. Weights can be assigned to rules to add a degree of uncertainty and modifiers of rules can be used to overcome ambiguity among concepts. RUBRIC has been shown to be as good as traditional information retrieval for finding relevant documents but better than traditional methods for finding marginally relevant documents, in other words it tends to have higher *recall* [9,13]. The greatest contribution of RUBRIC is that it allows the development of concepts to be done completely independently of a data source. This critical aspect of the separation of concept from data is described later.

3. A MODEL FOR A NODE IN A HETEROGENOUS NETWORK

As previously stated, the requirements for information will take commanders outside their systems. They will go to nodes on an underlying network. Figure 1 shows a schematic of a node.

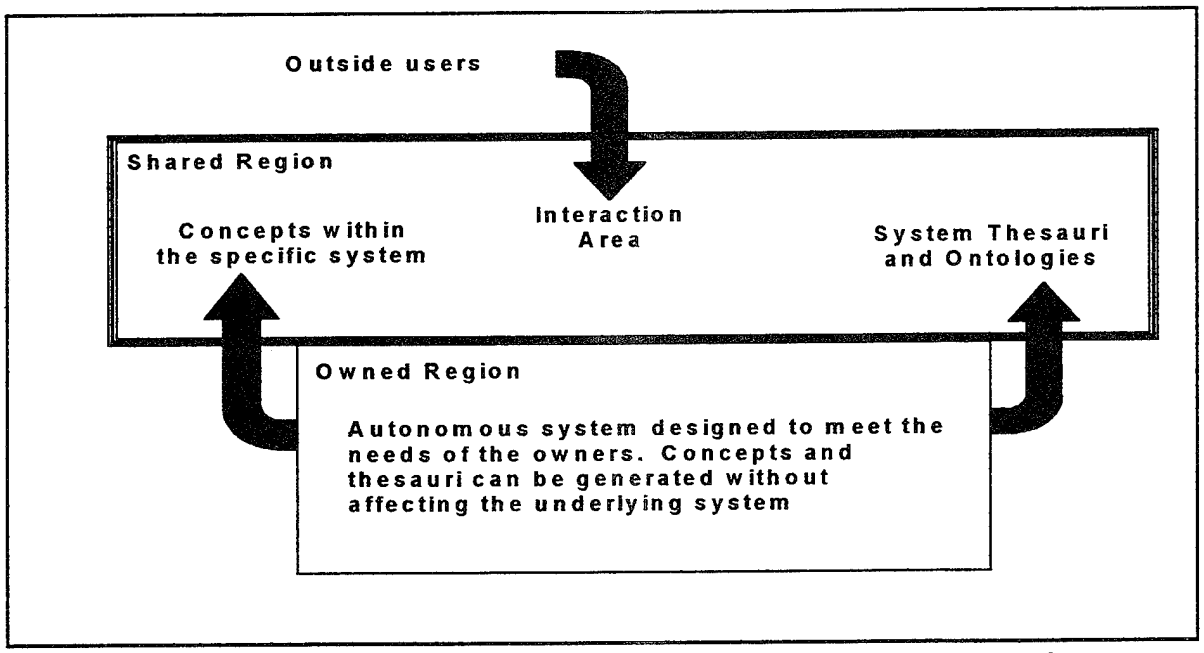


Figure 1: Schematic of a node in a heterogenous network

The reference to "Outside Users" represents the network resources such as backbones, gateways, bridges, protocols, etc. It also represent the Senior Decision Maker who has a query for information. The node itself consists of two separate regions. The Shared Region is the entry point for the decision maker. This region contains information about the system. The Shared Region represents what the system owners wish to make available to any commander. The Owned Region is private. Outside users only have access through very controlled processes. The Owned Region is designed, developed, and maintained strictly to support the owners. It is autonomous from the rest of the network.

The Shared Region is divided into three areas. The first area is the Concepts area. This is a knowledge base which represents the concepts contained within the owned system. The arrow goes outside the Owned Region because the sharable knowledge base is derived from the owned system but is not part of the system. In other words, the concepts must be derived and maintained without effecting or requiring changes to the underlying system. Another section of the Shared Region contains the system thesaurus and ontology. The thesaurus and ontology, like the concepts, are derived from the owned system. The thesauri and ontology may be directly based on the owned system or they may be a standard set, i.e., a domain related thesaurus/ontology of terrorist terms provided by a regulatory agency. It is possible to have several domain specific thesauri and ontologies in the Shared Region. The third area is the Interaction area. This is the only area the outside decision maker deals with. A more detailed look at these areas, as well as the process a query undergoes is shown in Figure 2.

The query enters the node to get processed (Step 1). The Interaction area maintains a record of previous queries it has serviced. Previous queries each have indicators that reflect their success. This indicator is a result of feedback from previous users. Several articles have

described the need for user feedback on query success yet none have indicated that the feedback should also be reflected in the servicing database, in this case within the Interaction Area. Previous results provide the decision maker an opportunity to decide if investigating the database further is warranted. If it is, (Step 2) the query moves to the system thesaurus. The query brings with it a portion of its "home" thesaurus. This thesaurus is merged with the system thesaurus. By converting query terms into thesaurus terms the query is transfigured into terms the system is more likely to understand. (Step 3) The query is sent to the concepts area where it is compared against concepts supported by the system. Once concepts are identified, (Step 4) they are passed into the Owned Region for retrieval. (Step 5) The retrieved documents are sent to the outside user. And finally, (Step 6) the user evaluates the success of the retrieval and sends feedback to the Interaction area of the servicing node.

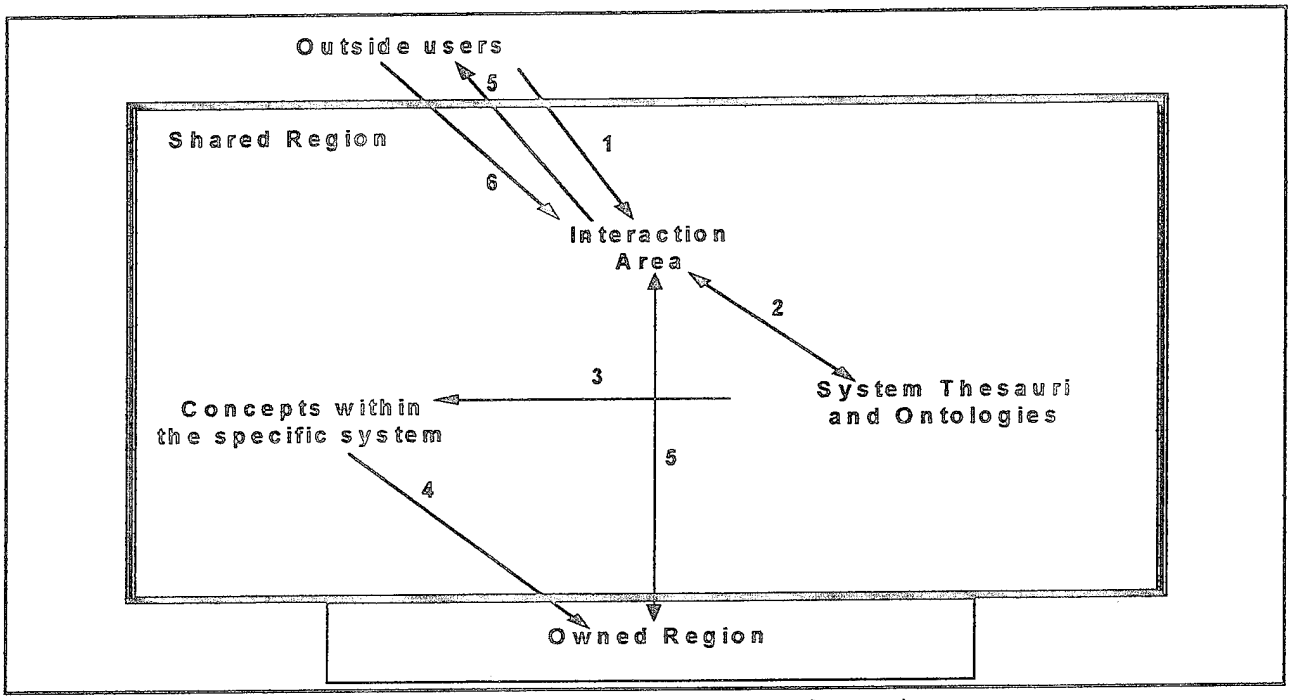


Figure 2: Query process through the Shared Region

4. MODEL VALIDITY BASED ON CURRENT RESEARCH EFFORTS

Each phase of the query process in the Shared Region is supported by current research. The underlying communication support is clearly recognized through the expansion of computer networks. In addition to ensuring proper permissions and protocols, the Interaction area maintains a usage history. It records the use of the Shared Region and the results of that use. This history has a key role for the decision maker. As part of step 1, the history record offers the decision maker an opportunity to decide to not use that database based on a statistical evaluation of previous use. While not explicitly stated, the user's system history is being updated to reflect a potential data source. If the user decides not to use this particular data source, his history file is annotated to reflect non-usage as well. However, even if not used (or subsequently found to be not useful) it is important for the decision maker to be able to remember where he/she has looked and how he/she got there (i.e., the communication path).

In step 2, queries are broadened by thesauri. Chen, Lynch et.al [6] used a blackboard design and a spreading-activation algorithm to traverse multiple thesauri within a single system. Their system used thesauri based on the Mosaic database, the ACM Computing Review Classification System (computing literature classified by the Association for Computing Machinery) and the Library of Congress Subject Heading thesaurus (specifically the general computing terms as selected and categorized by the Library of Congress). In an interactive mode the user browses among the thesauri. Search terms are recommended by the system and those selected are activated. In an activation mode thesauri would be activated based on user provided parameters. Weights are assigned among related terms within each thesaurus. An imbedded transformation function is used to ensure the activation of related terms stops after a suitable number of terms are found.

The net effect is that the various thesauri are merged to meet the needs of the user. The merged thesaurus is available through the blackboard. It is assumed that the end result is lost after the query has been executed. In the model, the query is not completed until the outside user sends back a measure of the success of the query. It is easy to extend that measure to be incorporated in an evaluation of the weighing factors used within the thesaurus.

Another method for thesaurus merging has been done by Knight and Luk [12]. They merged thesauri and ontologies which were based on English and Spanish. They used several different sources such as online dictionaries, semantic networks and bilingual references. Much of their merging was based on definition matching and lexical usage among the different sources. They also used a hierarchy matching algorithm. Combining the two algorithms gave very good results (96% accuracy). Once this was done, they developed the bilingual ontology. They used mappings between words based on the Collin's bilingual dictionary. They also used mappings between English words and their ontological entities. Ambiguities were flagged by the system for human correction and verification.

Both of the above approaches were deemed successful. But there is a big difference in the underlying premise of their work. Chen's [6] thesaurus is temporary to meet the requirement specified by a query. Knight and Luk [12] develop a thesaurus which becomes a permanent feature of the system. Both have their use in the model. The approach taken by Knight and Luk builds the system thesauri based on the document base. It is the approach implied by the arrow in figure 1 leading from the Owned Region to the system thesauri. Chen's approach is used within the Shared Region.

Once the query has been processed into terms the system uses, it is sent to the Concepts area for step 3. The Concepts area is probably the hardest to quantify as it deals with the concepts within the database. Gaines and Shaw [8] detail a set of tools which can be useful for eliciting knowledge from experts in putting together a system. While some are automated, the tools predominantly establish a procedural approach to elicit knowledge from human experts. However, they emphasize one point, that is "there is no royal road to ... system development. Understanding and communicating expertise are not easy tasks" [page 13].

Basu [3] has strongly argued against any type of monolithic knowledge base as the basis

for interoperability. He has developed a model of a Multiple Knowledge Based System (MKBS). He details concepts of a local problem, delegated problem and a shared problem. The local problem is solved internal to a system. A delegated problem is sent to another member of the MKBS who works on it and returns an answer. A shared problem is one of interaction between systems at different times to arrive at a solution. The biggest aspect of his work, and how it relates here, is that it describes a procedure for translating a rule from one knowledge base to another even (or especially) when predicates do not match. There is one drawback to his approach. He has argued against a monolithic knowledge base but requires all the members of the MKBS to use the same language. As previously discussed, a single language is not realistic. Heterogeneous systems will have to rely on translators to bridge the language differences. Over time there will probably be a convergence to a set of common requirements for interoperability.

Ginsberg [10] goes one step further. He describes the thesaurus as the complete source for all concepts within the database. A lattice structure links the thesaurus directly to documents within the database. By extensions to related words and various measure of relevance, documents become clustered to meet a query. He has successfully implemented his approach with a system called WorldView. In terms of the model, he has meshed the thesaurus area with the concept area. However, somewhat counter to the model, this approach maintains too strong a tie between what would be the shared area (thesaurus) and the Owned area (the documents). For this reason, RUBRIC, or a RUBRIC-like, system may be the best approach for the Concept area. RUBRIC allows the development of concepts independent of the database. If the user sends the query out of his/her system using RUBRIC than another system using RUBRIC should have little difficulty in combining and comparing concepts. While the use of the same conceptual language is beneficial, it is not required.

SIMS (Services for Information and Management for decision Systems) by Arens, Chee, Hsu and Knoblock [2] also supports the underlying premise of the Concept area in the model. The developers have been successful at integrating different Oracle databases using a LOOM knowledge base. They did not require a user to be familiar with how or where data were distributed. However they did require the user to be familiar with the domain. SIMS differs from other work in that its model provides the terms which the contents of the databases or other knowledge bases with which it interacts. The domain model of SIMS is not specific to the data sources and there may not be a direct mapping from the model concepts to data objects. This allows a great deal of flexibility in adding, deleting, and modifying the underlying data sources. SIMS preserves its independence from the data models. SIMS uses LOOM as its knowledge base.

LOOM consists of definitions, rules, facts and default rules. In LOOM a variable is a class and members (specific values) of the class are instances in the knowledge base. Instead of populating the knowledge base with all instances, it uses a LOOM Interface Module (LIM). The LIM takes the query from LOOM and does the reformulation into the query language of the underlying system. Database query results are passed to the LIM which reformulates them and gives them to LOOM as if they were instances generated by LOOM. LIM is oriented towards a single database, therefore some mechanism is needed to account for the individual databases. SIMS uses a planner which takes queries via the LIM, recognizes where the data elements are

which comprise the query and do the retrieval. SIMS has been used with the ORACLE database.

From the concept area, the query is sent to the database for retrieval of documents (step 4). In SIMS the interface to the database is part of the system. If the Concept area is based on RUBRIC a separate interface module may be needed. By keeping the concept area separate from the data sources it is possible to have access to a number of such interfaces. Thus data could come not only from different ORACLE databases as in SIMS but also from different databases which are a mixture of ORACLE, SYBASE, or INFORMIX, etc. This latter approach is more realistic since many of the legacy databases were established as stand-alone, single application systems [14,16].

Step 5 of the model is data retrieval. This is left as the process which is used by the underlying system. The final step is feedback (step 6). The decision maker's system will add the path to the node to its history file. Anyone who has dealt with some of the larger discovery services (gopher, Archie, Netfind, WAIS) has been faced with going through a session, come back a few days later, and realize they don't remember where they were, or they don't remember how to get to where they want to be. The feedback in this case is not feedback to the user but feedback from the user to the Shared Region. The user has expended time in the retrieval. If that retrieval was fruitful he informs the system. The next time a user comes looking for similar information, he can get a estimate based on past searches by others. It is also important to keep track of unfruitful searches. This way a commander can save time and resources if the estimate doesn't justify the commander conducting a further search. In any case, the final decision to conduct the search will reside with that commander.

5. CONCLUSION

The adage that "knowledge is power" has never been more true than in today's environment. With massive data collections and vast networks, the power is not in having the data but in the ability to go to where the data is and the ability to use it. The paper has described an information model which takes the information in heterogeneous data sources and provides that data to a commander. The commander, through the model, has access to information without needing to know the underlying composition of the data within that source. The model is able to adjust what the senior decision maker needs with what the data source can provide. At the same time, the data owner is free of collecting special data simply to meet the need of a potential user. Since the data owner and the data user are independent, the model offers a great deal of flexibility. Nodes which can provide data to a commander for one operation, need never be queried for a different type of operation. Consequently, the nodes within the network will constantly be shifting in terms of which operation they are actually supporting. This shifting is completely transparent to both data owners and the decision makers. Once the operation is underway, the nodes in support may change as new data is collected and other needs arise. The model enforces the ability to alter support without the need for complex network configuration management operations. It serves to bring the current operational environment closer to the seamless integration of systems required by the commander and senior decision makers for mission success.

REFERENCES:

[1] P. Anich, "Integrating Natural Language Processing and Information Retrieval in a Troubleshooting Help Desk," I*EEE Expert, 8(6), 1993, pp 9-18.*

[2] Y. Arens, C.Y. Chee, C.N. Hsu, C. Knoblock, "Retrieving and Integrating Data from Multiple Sources," *International Journal of Intelligent and Cooperative Information Systems," 2(2), 1993, pp 127-158.*

[3] A. Basu, "A Knowledge Representation Model for Multiuser Knowledge-Based Systems," *IEEE Transactions on Knowledge and Data Engineering, 5(2), 1993, pp 177-190.*

[4] D. Batty, "Thesaurus Construction and maintenance: A Survival Kit," *Database, 12(1), 1989, pp 13-20.*

[5] H. Chen, K. Lynch, "Automatic Construction of Networks of Concepts Characterizing Document Databases," *IEEE Transactions on Systems, Man and Cybernetics, 22(5), 1992, pp 885-902.*

[6] H. Chen, K. Lynch, K. Basu, T.D. Ng, "Generating, Integrating, and Activating Thesauri for Concept-Based Document Retrieval," *IEEE Expert, 8(2), 1993, pp 25-34.*

[7] C.J. Crouch, "An Approach to the Automatic Construction of Global Thesauri," *Information Processing and Management, 26(5), 1990, pp 629-640.*

[8] B.R. Gaines, M.L.G. Shaw, "Eliciting knowledge and transferring it effectively to a Knowledge-Based System," *IEEE Transactions on Knowledge and Data Engineering, 5(1), 1993, pp 4-15.*

[9] F. Gey, W. Cahn, "Comparing Vector Space Retrieval with the RUBRIC Expert System," *SIGIR Forum, 23(1-2), 1988, pp 5-15.*

[10] A. Ginsberg, "A Unified Approach to Automatic Indexing and Information Retrieval," *IEEE Expert, 8(5), 1993, pp* 46-56.

[11] U. Guntzer, G. Juttner, G. Seegmuller, F. Sarre, "Automatic Thesaurus Construction by Machine Learning from Retrieval Sessions," *Information Processing and Management, 25(3), 1989, pp 265-273.*

[12] K. Knight, S. Luk, "Building a Large-Scale Knowledge Base for Machine Translation," In *AAAI-94.*

[13] B.P. McCune, R.M. Tong, J.F. Dean, D.G. Shapiro, "RUBRIC: A System for Rule-Based Information Retrieval," *IEEE Transactions on Software Engineering SE-11(9), 1985, pp 939-945.*

[14] R. Neches, "Knowledge Sharing in Integrated User Support Environments: Applications, Frameworks, and Infrastructure," *KB&KS:Proceedings of the International Conference on Building and Sharing of Very Large-Scale Knowledge Bases '93, Tokyo, Japan, Dec. 1993, pp167-176.*

[15] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley Publishing Company, Reading, MA., 1989.

[16] W. Swartout, R. Neches, R. Patil, "Knowledge Sharing: Prospects and Challenges," *KB&KS:Proceedings of the International Conference on Building and Sharing of Very Large-Scale Knowledge Bases '93, Tokyo, Japan, Dec. 1993, pp 95-102.*

Author Biographies:

Anthony Ruocco is a Major in the U.S. Army and PhD candidate in Information Technology at George Mason University, Fairfax VA. As a member of the Operations Directorate (J3), U.S. European Command, Stuttgart, Germany, he worked extensively with major C2 systems in support of world-wide unit deployments from 1990-1993. As the directorate System Integration Officer he established functional requirements for new system development and integration of present C2 systems. He is a member of the Army Acquisition Corps, with a specialty in Automated Data Processing. He can be reached at 3912 Bokel Drive, Chantilly, VA, 22021. His number is (703) 378-6126 and email address is aruocco@gmu.edu

Dr. Ophir Frieder joined the Computer Science Faculty at George Mason University after three years in the Applied Research Area of Bellcore. In 1993, Dr. Frieder was named as a recipient of the International Information Science Foundation Award from Japan and the NSF National Young Investigator Award. Dr. Frieder's research interests include parallel and distributed database and information retrieval systems and biological and medical data processing architectures. He currently serves as an Associate Editor in Chief of IEEE Software and is a member of Phi Beta Kappa and the ASEE, and a Senior Member of the IEEE. He can be reached at The School of Information Technology and Engineering, George Mason University, Fairfax VA 22030-4444. His number is (703) 993-1540 and email address is ophir@cs.gmu.edu

IMPLEMENTING STANDING REQUESTS FOR INFORMATION OVER DISTRIBUTED HETEROGENEOUS DATA SOURCES

DR. KENNETH SMITH
PATRICIA L. CARBONE
MICHAEL S. V. TURNER

THE MITRE CORPORATION

Abstract. The objective of the MITRE Intelligent Software Agents (ISA) project is to create intelligent agents that automatically create and monitor what are called "standing requests for information" (SRIs) which act as the commander's surrogate to detect critical changes in the state of the world as reflected in various available data sources. Planners require knowledge of crucial changes in the relevant world to make necessary mid-stream adjustments in the course of action during a crisis. However, current data sources are not coupled with planners in a proactive manner; decisions are made on relatively "old" data because updates have not been propagated. This decoupling is complicated by the fact that relevant data sources are voluminous, remote, unfamiliar and often inconsistently formatted, and that many changes to data sources are irrelevant to the needs of the planner. Without assistance, users could be overwhelmed by the volume of information in the data sources, with the task of transforming updates into a familiar view for their task at hand, and with determining exactly which updates are important before they age beyond usefulness. SRIs, in the form of *intelligent agent* technology, are designed to provide timely and continual reporting of only those changes that matter, on a single consistent view corresponding to the situation relevant to the planning task. We use metadata mappings to combine many heterogeneous data sources into a single federated view called a *situation*. SRIs are then defined over the situation, and changes to the data sources are transformed into changes on the situation view. Our current focus in this project is the implementation of SRIs in a manner that reduces information overload by two levels of data filtering, increases access by caching relevant materialized portions of the situation locally, and utilizes rapidly emerging COTS technology to the fullest advantage, including active DBMSs replication servers to proactively forward changes, and metadata repositories to assist in data reduction and transformation into a federated view. This paper presents three potential architectures for implementing SRIs: full materialization, demand query evaluation, and partial materialization through caching. It gives an analysis of these approaches including their expected impact on efficiency and data consistency requirements. Finally, we describe our prototype caching architecture in more detail.

1. INTRODUCTION

Currently, there is a critical need in the military to provide interoperability among and access to database systems that are maintained within different branches of the military in order to support integrated command and control decision support. Because of the large number of military and humanitarian operations occurring around the world that involve a joint task force, access to each branch's data by a single joint task force commander or centralized focal point is becoming increasingly important. This is evidenced by the implementation of the military's Global Command and Control System (GCCS), which is being developed to allow interoperability between each of the military branches at a Commander-in-Chief-level command center.

Increased access to more data is not necessarily the end goal, as large amounts of data will soon overwhelm the user. In order for an integrated system to become a true decision support system, the system must be able to consolidate and synthesize the data to an understandable level of abstraction for easy understanding by the user. A new technology, called *intelligent software agents* (ISAs), will enable the system to automatically monitor situation data to support

development of a course of action (COA) plan (performed by the planner segment of the system). In essence, ISAs provide automated generation and monitoring of *standing requests for information* (SRIs). [NOTE: An SRI describes the situations that the commander wants to be notified of, should they occur.] One of the technical challenges for "intelligent" agents is enabling these agents to automatically define observables that need to be monitored, monitor the situation, present information to the user relevant to current decisions, and reassess the plan based on information presented. A major challenge for agents alone is to monitor heterogeneous data sources through a single federated view, or a situation. Through use of ISAs, the integrated system will provide true decision support required by the commander.

It is important to note that the COA plan guides information collection and fusion performed by the ISAs, and SRIs indicate the information necessary for "plan execution monitoring." It is the plan that provides the context for what would otherwise devolve into an overwhelming flow of data back to the decision maker. These concepts are common over all the services, since working from a plan or a course of action is common to all. In addition, it is important to note that the SRIs may be displayed to and modified by the users.

2. BACKGROUND

An integrated command and control decision support system should provide such functionality as situation assessment, planning, tasking, determining what support is needed for the plans and tasks, execution of the plans, and situation monitoring. When the planning system and various data sources are not integrated, the staff must try to monitor SRIs manually, accessing the various data sources and mentally integrating the data to see if the commander should be notified that a change in the plan may be necessary. This procedure will become more of a problem as the probability of data overload becomes more of a reality. The data are extremely dynamic, they are incomplete, data can be contradictory between two sources, and the user can be uncertain about the relevance of the data to the plans. Therefore, decision support should manage the information gathering for the user, changing the data into information by performing high-level information fusion and explaining the relevance of the data to the plans.

Project Objective

The objective of this project is to reduce the amount of data being sent to a user and to demonstrate improved decision support and planning through use of automated situation monitoring via *intelligent software agents* .

What is an Intelligent Agent?

An intelligent agent (IA) is a robust, autonomous process that communicates with other entities to gather information and make decisions. IAs typically have two basic characteristics. One is the intelligent behavior, typified by the packaging of a set of artificial intelligence techniques such as knowledge representation and reasoning about a particular domain and a set of background knowledge. An IA exhibits intelligent behavior by being able to reason with high level abstractions (*e.g.*, plans or SRIs) while operating with low level information (*e.g.*, triggers) in target data sources. The other characteristic is the agency aspect, in that the process operates independently, continuously, and asynchronously. IAs communicate with other entities, agents, and processes using some sort of agent communication language. [CACM94] is a good reference for intelligent agent definitions as well as descriptions of ongoing intelligent agent research. There are numerous types of IAs being developed, including information filtering agents, information acquisition agents, cooperative scheduling agents, multitask execution agents, and cooperative problem

solving agents. These different types of agents vary in terms of the minimum amount of intelligence and distributed agent communication that is required to address a set of goals.

For our effort, ISAs will be information acquisition agents to be able to actively monitor a situation by providing automated generation and monitoring of SRIs. The ISAs will automatically monitor the situation, in the context of a course of action plan, and present relevant information to a user and/or a planner system.

Specifically, each ISA in our project will:

- Analyze the COA plan to determine the observables to be monitored;
- Construct triggers that define constraints on those observables and place them in the database(s);
- Reassess the plan based on information presented (*i.e.*, data returned by the databases);
- Alert the user to critical changes in the situation; and
- Explain the relevance of the returned information to the plan.

Our project is currently in the first year of development. During 1995, we will be concentrating on developing agents that monitor the situation in the form of SRIs. In this time period, the user will provide the "intelligent" behavior by defining SRIs on the situation. During 1996, we plan to add knowledge representation and reasoning so that the SRIs can be generated and modified automatically. The next section describes the issues involved in developing agents to monitor heterogeneous data sources.

3. IMPLEMENTATION OF SRIs

During 1995, we are concentrating on developing the agency aspect of the ISA. Our work has focused on how to monitor the databases that have data relevant to the current situation, given that relevant data sources are voluminous, remote, and unfamiliar to the user. In addition, we want only to alert the user to relevant changes in the data, not to all changes that may not be pertinent to the plan. We are using the active database and replication server technology to implement the SRIs that will in turn notify the user when a critical event has occurred. One of the key technical issues of the ISAs was how to use the active databases to monitor the current situation. We are developing techniques for defining triggers on a database view, rather than on base tables. A database view is a virtual table whose contents are defined by a standing query. The importance of views is that they allow database administrators to provide views of the data which are customized to the needs of different user groups. These views can span multiple physical databases and can shield the user from the peculiarities of each individual database.

Problem Description

The problem we are trying to address is the following. Given a plan and a set of databases containing continuously updated state-of-the-world information impacting the plan, we want to devise an alerting architecture to connect the databases with the plan so that the plan is constantly based on timely information. The alerting architecture must include four major interconnected parts (see Figure 1):

- Component databases. Component databases are constantly being updated with status information and are distributed across a network. Our current emphasis is on commercial database management system products, although databases could include flat files and legacy databases. We also assume that the databases are autonomous and are not clients of the machine on which the situation is being maintained.

- Situation. Since component databases do not individually present the entire picture needed by the planner, a "situation" is defined which combines relevant information from many heterogeneous data sources into a composite picture. The situation provides a schema against which standing requests for information (SRI) queries can be posed.

- Standing requests for information (SRIs). SRIs are binary threshold queries which identify boundaries of values crucial to the plan, such as "is there currently a large runway open in city x?" Crossing the threshold is assumed to remove or make available a plan state for use in possible plans, signaling an appropriate time to replan.

- Planner. This is the planning system for the particular domain (e.g., transportation planning). We assume that an agent will be able to extract the critical SRIs from the plan and that an interface is defined by which the planner is notified when a threshold is crossed.

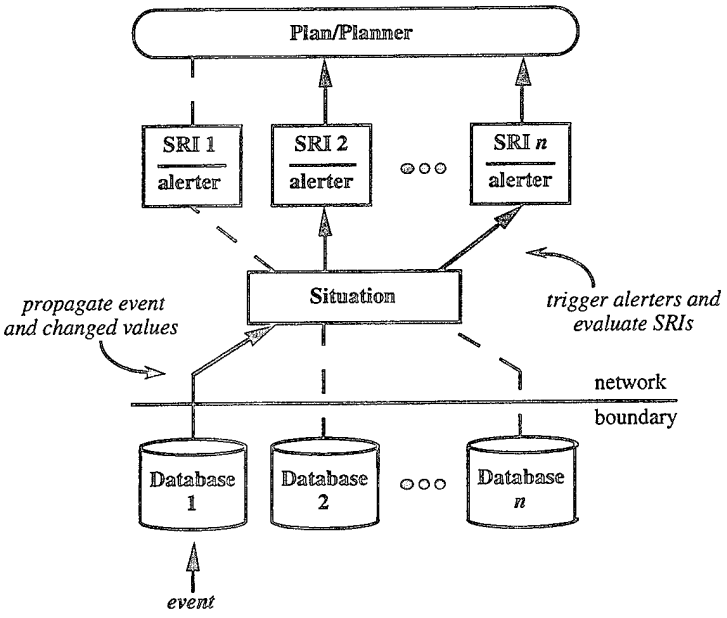


Figure 1. Event Propagation

Initially, the situation is defined (via interaction among database administrators and planning experts) and can be written as a relational schema (although whether or not the situation is actually materialized is a question we will address). The situation should be broad enough to support the anticipated SRI queries. Then SRI queries are written against that schema by the planning experts. The system will then "compile" these SRIs into a suitable alerting mechanism which can detect relevant change events and evaluate SRIs based on the latest component database values.

In order to implement standing requests for information, we need to be able to define triggers over views. Many current systems (e.g., Oracle, Sybase, Illustra) support defining triggers over base tables; however, they do not support triggers defined in terms of a view. We are applying previous results in database consistency management and incremental maintenance of materialized

views to the problem of defining triggers against a view and automatically generating triggers against base tables that can be processed by commercial database management systems.

These parts interact at runtime as follows and as shown in Figure 1. As an event alters the relevant part of a component database, the event and content of this change are propagated to the situation derived from those databases. When the altering architecture detects that the part of the situation pertaining to a particular SRI has changed, that SRI is reevaluated against the situation.

Preliminary Technical Issues

There are two preliminary technical issues to address which focus the type of architecture that will be developed and implemented. The first issue deals with whether or not the situation is materialized in a database somewhere. The second issue deals with how the SRIs should be evaluated. These issues are described below.

Materialization of the view. The situation definitely exists in the mind of the person defining the SRIs, since SRIs are formulated against the situation. However, we must decide whether the situation is actually materialized or not. One approach is to treat the situation like an unmaterialized view. The situation would not exist as a database populated with tuples which could be queried. Instead, SRI queries against the situation would be transformed into queries against the underlying component databases through *query modification*, a common implementation strategy for database views. In this case, the actual implementation would appear different from the conceptual architecture above, since we would be querying a *virtual situation.*

A second approach is to materialize the situation. In this case the situation is implemented as an actual database with schema and tuples, using some DBMS product. Every update on a component database which affects the situation is propagated and transformed into a corresponding update on the situation. Alerters are implemented using triggers on the situation database to detect updates from the component databases. The resulting SRI queries would be directed to the database and answered using its query language.

We have chosen the unmaterialized situation approach for the following reasons:

- Since our main task is responding to SRI changes, maintaining the entire situation is overkill and could slow response time.
- Middleware exists to support the non-materialized approach. This could greatly ease our implementation.

Evaluation strategy for the SRIs. The concept of a *standing* request for information reflects a desire to be proactively notified of all changes. This is the data-driven (or eager evaluation, or forward-chaining) approach in which updates are immediately propagated throughout the system until a quiescent consistent state of the database (situation) is reached. The database can then be queried with confidence that the resident data reflects the most recent state of the world.

The data-driven approach can be simulated by the demand-driven (or lazy evaluation, or backward chaining) approach in which updates are not propagated. When a query (demand) is issued, exactly those parts of the database which are required to answer the query are first reconciled with all recent changes, then the query is answered. If each SRI is demand evaluated at a regular interval *i*, the answer is never more than *i* out of date. If that is within acceptable tolerances for the planner, this approach can be used as well. A situation in which this approach would be useful is updating a human readable real-time display with a refresh interval of *i*.

Unless *i* is large, the data-driven approach is probably more efficient. In addition, this approach gives the most up-to-date answer possible, which is most consistent with SRI semantics. We will favor the data-driven strategy.

Solution Strategies

We investigated two major approaches to implement the prototype system which were consistent with the unmaterialized, data-driven strategy. We discuss them here.

Approach 1: Basic Query Modification. In this approach, the situation is unmaterialized and change events are propagated in a data-driven fashion, but the actual changes are evaluated in a demand-driven fashion by decomposing the SRI query. Initially, SRIs are compiled into triggers set on the component databases. A trigger is set for every update event which one or more SRIs need to be aware of. Since the situation is a view, a mapping must be performed between the updates required to trigger an SRI and the corresponding updates on the component database(s).

When a trigger fires, it notifies its SRIs that their thresholds may have been crossed. It is impossible to know for certain if reevaluation is needed, because an SRI may act on the logical "and" of two conditions, only one of which is local to the trigger. So trigger firing is necessary but not sufficient for an SRI threshold change, and extra evaluations can result. Once notified, the SRI must be evaluated. The SRI query is transformed into queries on the component databases, and the results are combined using the view expression which defines the situation. The result answers the SRI, and a replanning recommendation may be given to the planner. Note that crossing the network boundary between the component database machines and the situation machine must happen once to propagate events and twice more to send the modified queries and receive their results.

SRI evaluation in this approach may be accomplished by using COTS middleware which would perform the query modification automatically. There may be an added complexity if heterogeneous rule (trigger) systems are used on the various component machines.

Approach 2: Situation Cache (SC). In this approach, the situation is also unmaterialized, but a cache on the situation machine is introduced containing partially processed portions of the situation relevant to existing SRIs. Change events are propagated in a data-driven fashion to the situation cache, as are the actual changes. Triggers on the cache notify the appropriate SRIs, and SRIs are then demand-evaluated against the cache (by these triggers). (See Figure 2.)

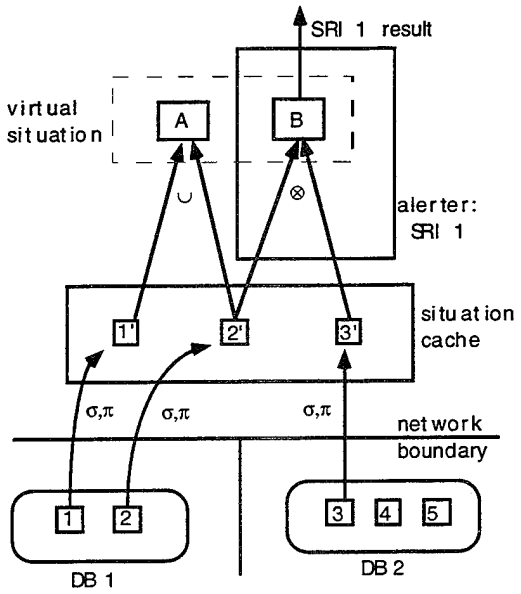


Figure 2. Approach Using a Situation Cache

Initially, a component-to-cache and cache-to-situation mapping is designed. Usually, the first mapping does unary relational operations such as projects and selects, and the second does multi-relational operations such as joins and unions. Then SRI's are compiled into triggers on the cache which detect updates to the cache relevant to one or more SRIs, and will perform the second mapping for each SRI. Since the cache is local to the SRIs, evaluating the SRIs does not cross the network boundary as in Approach 1, and if it were possible to detect complex events (such as "A and B are updated"), it would be possible to only evaluate SRIs when they really change.

Two strategies can be used to keep the cache updated. If a COTS replication server is available, it is likely it can be used to automatically maintain exactly the relevant portions of the component databases as replicates on the situation server, perhaps even performing the selects and projects automatically. If no replication server is available (or compatible), the SRIs must be compiled into triggers on the component databases as well. These triggers propagate both the change event and the changed tuples across the network boundary to the cache, and perform the updates to the cache. Note that only one crossing of the network boundary is required in either variation.

Strategy Selection

Under query modification, the procedure to activate an SRI (and generate a notification) involves all of the following steps:

1. An event detected at a source database is determined to possibly affect an SRI. As mentioned above, we cannot know beforehand for certain if an event will cause an SRI to need to be evaluated.
2. The event is propagated across the network to the SRI agent.
3. Sub-queries on source databases corresponding to the SRI are issued back across the network to the sources.
4. The answers to each subquery are returned across the network to the SRI site.
5. Heterogeneity transformations are performed on the result of each sub-query to translate source data into a format compatible with the situation.

6 These results are combined and the SRI is evaluated. If true, a notification is issued.

Under the situation cache approach, the following steps are required to activate an SRI:

1 An event detected at a source database is determined to possibly affect an SRI.
2 The event is propagated across the network to the SRI agent. The raw changes associated with that single event are simultaneously propagated to the situation cache.
3 Heterogeneity transforms are applied to these changes.
4 The SRI is evaluated against the cache, if true a notification is issued.

Note that much less data crosses the network boundary and much fewer (one instead of three) network boundary crossings are performed in the situation cache method. Although the price is the maintenance of a cache at the SRI site, we feel that this will be justified by improved performance (faster notifications). Therefore, we have chosen the situation cache approach.

4. ARCHITECTURE

Figure 3 shows the system architecture for 1995. For this year, the user is acting as the "intelligent" part of the ISA, in that the user will define and modify all SRIs. The "agent" part of the ISA will be the automated SRI and perform the information acquisition.

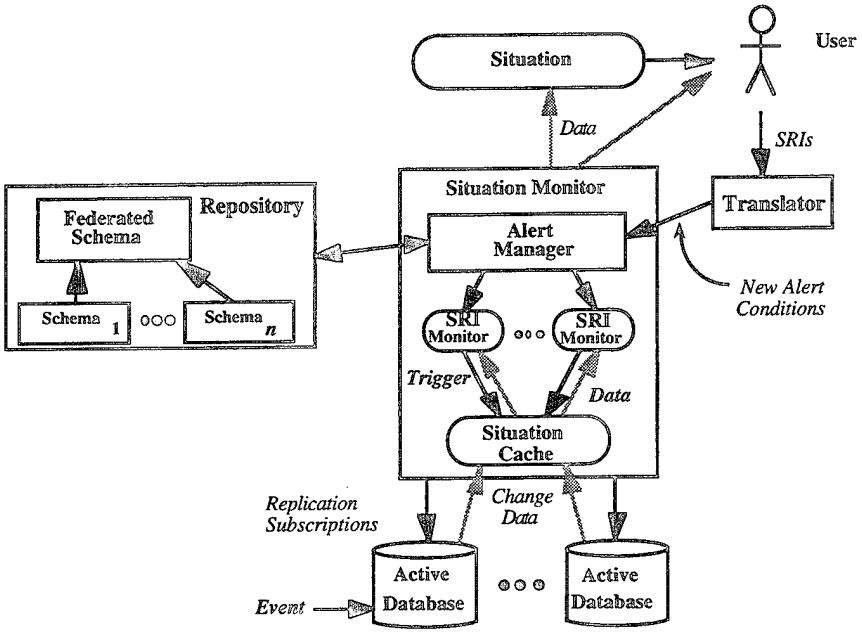


Figure 3. 1995 ISA System Architecture

A user defines a SRI in the context of a plan that has been defined. That SRI is translated into a set of alert conditions. The Alert Manager in the Situation Monitor works with the Repository to define a set of SRI Monitors that are compliant with the alert conditions. The Repository contains metadata about each of the component databases in the system and their schemas as well as the federated schema that supports the situation description. The SRI Monitors create triggers that are placed on the situation cache. In addition, a set of replication subscriptions are created and sent to the component databases. As events occur and the component databases are updated, the replication servers send the pertinent situation data to the Situation Monitor's Situation Cache.

In 1996, we intend to have a complete implementation of the ISAs (both intelligence and agency). Figure 4 shows the architecture for 1996, allowing the automatic creation and modification of the defined SRIs.

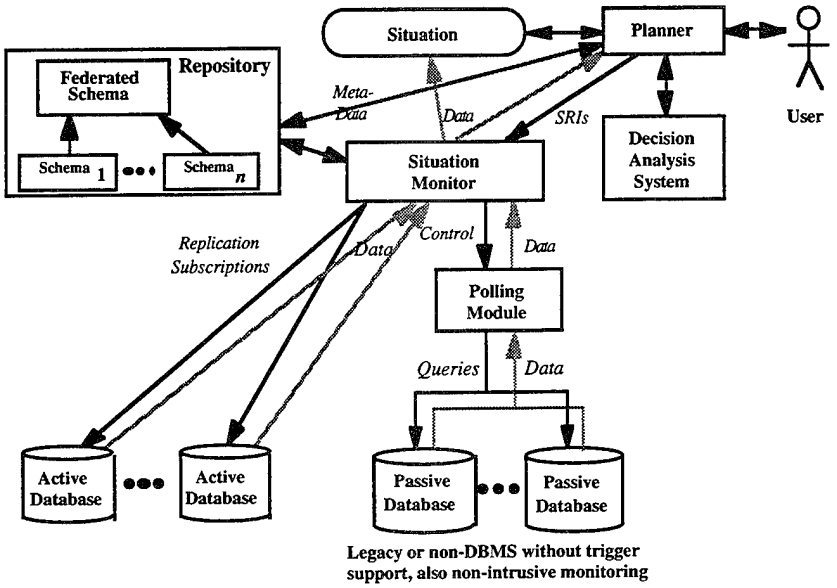


Figure 4. 1996 ISA System Architecture

Figure 4 shows the addition of a Decision Analysis System. The user will interact with the planning system to develop the plan. When the plan is complete, the Decision Analysis System will transform the plan from its internal representation into a Bayesian network, where the nodes in the network represent decision points and possible outcomes. The analysis of these nodes will cause the generation of SRIs which are sent to the Situation Monitor (as described above and represented in Figure 3). In addition to sending subscriptions to component database replication servers, a polling module can be implemented that polls passive legacy databases for pertinent data to be included in the Situation Cache. When pertinent data is identified from any of the sources, the data are sent to the situation, and the probabilities in the Bayesian network are updated. Suggestions can be made to the user as to how the plan should be changed, or the plan can be changed automatically. In either case, when the plan is modified, the Decision Analysis System will look at the new plan and either modify, delete, or create SRIs to conform with the new plan.

5. EXAMPLE

We envision great potential for our technology in many domains, including transportation logistics, air and cruise missile mission planning, and others. We have defined an operational scenario in the area of transportation logistics that has many characteristics that require situation monitoring. The scenario involves airbase capacity (*i.e.*, Maximum On Ground (MOG)). MOG is concerned with such aspects as parking space for planes, storage space for cargo, and loader equipment, as well as logistical considerations like food and sleeping quarters for crews, maintenance personnel, and fuel, among others.

Scenario

The following is a brief overview of the scenario we developed (depicted in Figure 5). A humanitarian relief effort is transporting supplies to Africa from the US via Spain (*i.e.*, 15 pallets of supplies need to be shipped from origin to destination). A plan was formulated and is currently still valid. The plan calls for three US C-130s to fly from AFB Alpha [US] to AFB Beta [Spain] (and back); transfer the cargo to two arriving NATO C-141s (with additional supplies) which will fly from AFB Beta [Spain] to AFB Delta [Africa] (and back).

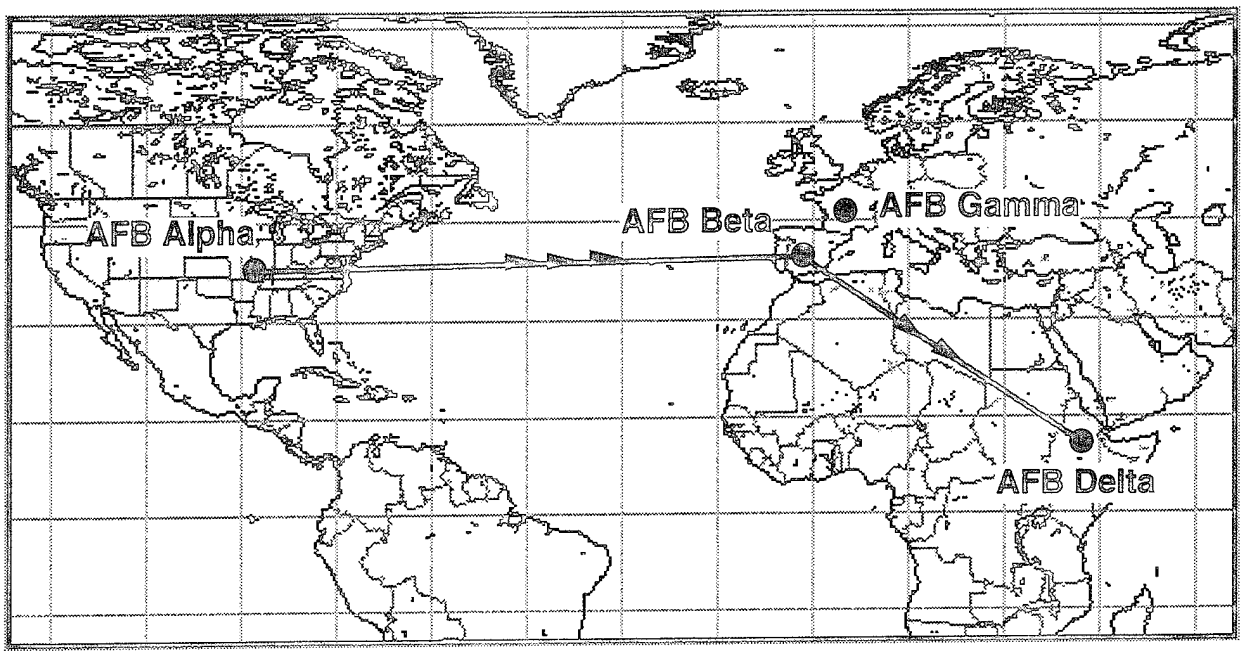


Figure 5. Transportation Logistics Scenario

The following is a partial list of unplanned events that could impact the current plan:

- Fuel Supply Truck delayed for either AFB Delta or Beta.
- Unexpected plane(s) land at either AFB Delta or Beta.
- Plane(s) need unexpected maintenance at either AFB Delta or Beta.
- A C141 arrives late at AFB Beta.
- A C141 needs maintenance at either AFB Delta or Beta (causing C130(s) to go to Delta).
- A C130 needs maintenance at either AFB Delta or Beta.
- Cargo transfer delay causing excessive storage needs at either AFB Delta or Beta.
- Emergency closure of either AFB Delta or Beta.
- Unexpected weather at either AFB Delta or Beta.
- Refuel Truck needs maintenance at either AFB Delta or Beta.
- Forklift needs maintenance at either AFB Delta or Beta.
- Loader needs maintenance at either AFB Delta or Beta.

Use of Intelligent Agents

Our system could be manually implemented if humans could accurately monitor multiple situations. In other words, the group of people would have to consistently translate high level requests for information into database-specific alerts; monitor hundreds of databases while knowing the interrelations of the data; and have a centralized group consolidating and translating the data into a common format. However, intelligent software agents are far more efficient, accurate, and more

cost effective at situation monitoring. In 1995, the system will look similar to that depicted in Figure 6. The user will develop a plan, and he will also use a SRI definition tool to define the SRIs for the plan. Using the previously described architecture, the system will place replication subscriptions in the component databases. When pertinent data is received, an alert will be displayed to the user who will in turn decide how to modify the plan and define new SRIs.

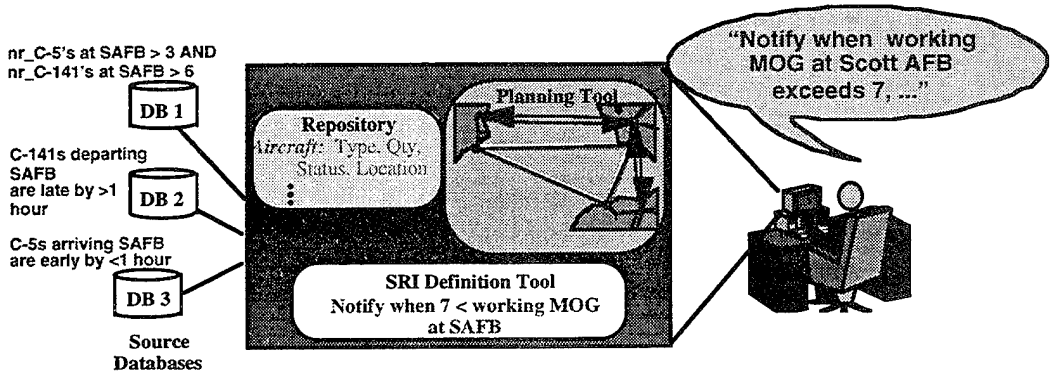


Figure 6. Depiction of ISA System in 1995

In 1996, the user will not be required to act as the intelligence for the agents, but instead will need only to look at the plan. (See Figure 7.) The intelligent agents will: automatically use the plan to determine the events to be monitored, send agents (triggers) to the various databases in the federation to monitor for those changes, synthesize the returned data and explain its relevance to the plan; replan as necessary; and either modify, delete or generate new agents to monitor for the new events. The users are then freed up to perform other duties or to make modifications to the alerts being generated automatically. Note, however, that the SRI Definition Tool will continue to be available so that users can modify or delete existing SRIs or can generate new SRIs that may be important.

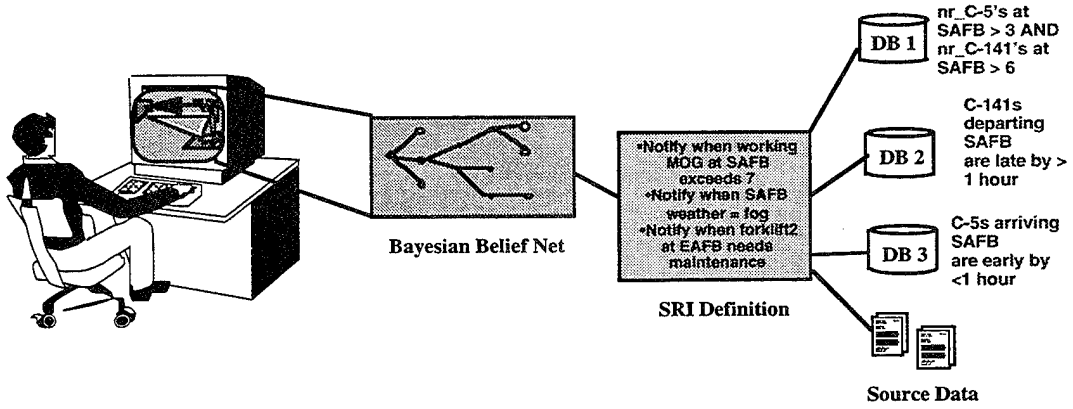


Figure 7. Depiction of ISA System in 1996

6. CONCLUSIONS

Our effort is performing leading edge research on the use of intelligent agents to perform situation monitoring. This year's work has focused on solving the problem of efficiently providing notifications of crucial changes to a virtual situation composed from a set of distributed

heterogeneous data sources. Our situation cache architecture provides a good tradeoff between a fully materialized architecture and a query modification approach. Network crossings are minimized and full materialization is avoided, while providing rapid accurate SRI notifications. We are currently developing a proof-of-concept demonstration of this technology using a real-world scenario. This year's research provides a solid foundation in developing the intelligent planning phase of our effort. By the end of FY96, the system will be able to automatically create agents to monitor situations identifying important planning information.

7. REFERENCES

[CACM94] Special Issue on Intelligent Agents, *Communications of the ACM*, Vol. 37, No. 7, July 1994.

8. AUTHORS

Dr. Smith received his Ph.D. in Computer Science from the University of Illinois at Urbana-Champaign. His work focused on data models for active database management applied to problems in engineering design. He continues to work in the areas of transaction management, active database technologies, concurrent engineering, and multilevel secure DBMS design with MITRE. Current address: The MITRE Corporation, 7525 Colshire Dr., MS Z231, McLean, VA, 22102. Telephone: (703)883-6115. Fax: (703)883-1397. E-mail: kps@mitre.org.

Ms. Carbone is the manager of the Intelligent Information Management and Exploitation Technology Area in the Advanced Information Technologies Center at MITRE. She is also the project leader for the Intelligent Software Agents project. She works in the areas of advanced database research and intelligent information management for the Navy, DMSO, and the ARPA/DISA Joint Program Office. Current address: The MITRE Corporation, 7525 Colshire Dr., MS Z459, McLean, VA, 22102. Telephone: (703)883-7963. Fax: (703)883-6435. E-mail: carbone@mitre.org.

Mr. Turner is a Senior Member of the Technical Staff in the Intelligent Information Management and Exploitation Technology Area in the Advanced Information Technologies Center at MITRE. He supports the Intelligent Software Agents project and other intelligent systems efforts. He is working towards an Sc.D. in Software and Systems from George Washington University. Current address: The MITRE Corporation, 7525 Colshire Dr., MS Z464, McLean, VA, 22102. Telephone: (703)883-6498. Fax: (703)883-6435. E-mail: mturner@mitre.org.

ENVOY: SUCCESSFUL MULTI-AGENCY DEPLOYMENT OF HETEROGENEOUS DATA ACCESS

Dan Stickel, Tom Hillman, Larry Safran, & Jerry Beersdorf
— Delfin Systems —

1. ABSTRACT

Envoy is an off-the-shelf application which provides easy, intuitive access to a wide variety of heterogeneous data sources (relational, legacy mainframe, and text repositories), making it possible to quickly retrieve all relevant information with just a single question. Envoy supports multimedia, and requires no changes to queried database systems. Users can run multiple questions concurrently, and manipulate results in an object-oriented environment that facilitates further manipulation and straightforward export into other software applications such as maps, timelines, link analysis tools, spreadsheets, and so forth. Envoy is currently being extended to work with Intelink both as an Intelink client and an Intelink server.

Envoy has been operationally deployed since 1993, and has undergone dramatic improvements in the intervening years. Current installations include USSOCOM, AIC, I Corps, III Corps, 18th Airborne, ONI, and other Government agencies, with more installations planned in the near future. Envoy has been identified as a DODIIS migration toolset, and has also been released as a commercial product distributed both domestically and internationally. Parties interested in obtaining a copy of Envoy at no cost (a limited number of starter licenses have been procured) should contact Commander Tom Cool at 301-669-5228.

2. INTRODUCTION & SHORT HISTORY

By the early 1990's, a wealth of on-line information was becoming available within the Department of Defense (DOD), but in such a bewildering variety of formats and access protocols that the average user had little chance of effectively utilizing these vast resources. Understanding the intricacies of various retrieval syntaxes, let alone discovering the correct database for a particular requirement, was beginning to present a real problem.

In 1992, the Joint National Intelligence Development Staff (JNIDS) awarded a multi-million dollar contract to Delfin Systems to create an off-the-shelf software application for providing intuitive access to multiple, heterogeneous databases. This application, named Envoy, would hide the complexities of heterogeneous data source access from the user, and would require *no* changes to the external systems being accessed (immediately

overcoming an entire class of potential obstacles). In order to provide a true end-to-end solution, it would incorporate several basic results manipulation and presentation capabilities, as well as support straightforward data export to standard analytical applications, such as spreadsheets, maps, and link analysis tools. Finally, while the application would include connections to several popular data sources right out of the box, it would also provide powerful facilities to create connections to additional sources through simple point & click.

In order to create a solution that potential users would immediately appreciate, and to create it in a timely fashion, JNIDS made the decision to use iterative, user-centric development. In early 1993, the first Envoy prototype was delivered to U.S. Special Operations Command (USSOCOM) ... and received a rather lukewarm response. The prototype generally functioned and appeared as planned, but users found the interface confusing, and the initial connection to a single external data source provided limited value. To make matters worse, the users were concurrently switched from familiar PCs to UNIX workstations, itself a major undertaking.

Fortunately, the user-centric, iterative development process functioned as designed, and over the next two years Envoy evolved to become the success it is today. First-time users are immediately impressed with its ease of use, and long-time users rely on it for their daily workload. Benefits include reduced training requirements, improved organizational response time, and greater confidence that all available information has been adequately reviewed. Envoy has been installed at multiple DOD commands and Government agencies, and a commercial version has been distributed both domestically and internationally.

Along the way, the Office of Naval Intelligence assumed responsibility for the program from the now-defunct JNIDS organization, and more importantly, Intelink sprang into existence (based upon HTML documents and web browsers) as the preferred method for data dissemination with DOD, providing yet another data source for Envoy to read, and also providing an opportunity for Envoy to act as a data mediator serving Intelink requests for information from more traditional database sources.

The remainder of this paper is partitioned into the following sections:

3. **Envoy Capabilities** — Describes delivered user capability, administration support, and Envoy's unfolding relationship with Intelink.

4. **Envoy Architecture** — Provides a brief overview of Envoy's client/server architecture, and its object-oriented connection classes.

5. **Deployment** — Reviews deployment locations and experiences, as well as providing contact information for obtaining a free copy of the software.

6. **Conclusions** — Summarizes overall Envoy experience and future directions.

7. **Authors** — Brief sketches of the authors and their contact information.

3. ENVOY CAPABILITIES

In brief, Envoy provides easy access to multiple, heterogeneous data sources. Users need not know specific data source syntaxes, or even which data sources contain applicable information. The Envoy user interface guides users through query specification with nary a hint of command line interaction, automatically determines the appropriate data sources, retrieves the desired information using the native language of the host database system, and presents returned results to the user in a variety of formats, including tables, text, imagery, and even video. Powerful options such as maps, timelines, network diagrams, and other analytical aids can assist with the evaluation and presentation of these returned results, via a notion of entity-centric computing.

A second method of interaction, currently under construction, allows external programs to submit data requests via CORBA (Common Object Request Broker Architecture) or HTTP (HyperText Transaction Protocol), so that Envoy can act as an advanced data mediator serving various user-specific front ends (for example, Intelink home pages).

In addition, Envoy includes powerful tools that enable local system administrators to connect to data sources beyond those automatically included in the package, such as local Oracle or Topic data repositories.

3.1 Easy Access to External Data Sources

Envoy provides an X-Windows graphical user interface, as well as CORBA and HTTP interfaces for servicing requests from external front ends. The X-Windows user interface has been refined via years of on-site interaction with sample users, with the result that simple questions are remarkably easy to ask, and more complex questions are supported with a depth of straightforward options.

Throughout the Envoy user interface, most selections include a *helper* button which assists the user in indicating his or her desires. This assistance may take the form of a list of legal formats for date entry, a list of known country names, a hierarchical display of available retrieval concepts (e.g., Facility, Manufacturing Facility, Military Manufacturing Facility), or even a map designed to enable users to restrict results to a specific region that they draw. This approach reflects the entire object-oriented nature of the system (i.e. operator overloading), and provides a visual cue that help is available without cluttering the display with different symbols that reflect the actual programming construct that will be used to satisfy the request for help.

Regardless of the particular user interface, while some questions involve the simple search for data containing certain user-entered keywords, many involve more complex relationships such as distance from a certain point, linkages with other entities, minimum sizes, and so forth. To handle these types of questions, Envoy provides a hierarchical list of relevant *concepts*, each with its own collection of associated *attributes*. Concepts and attributes are similar to tables and columns, but for their object-oriented nature that includes hierarchical subtypes and inheritance. Thus, if a user asks for all Facilities in a certain area, Envoy will know to return all general Facilities, including those that may be specialized subtypes such as Manufacturing Facilities or Distribution Facilities. Moreover, each concept normally has associated certain default constraints and associations which may be overridden if desired, but which simplify user interaction and generally tend to produce the desired results with a minimum of user direction.

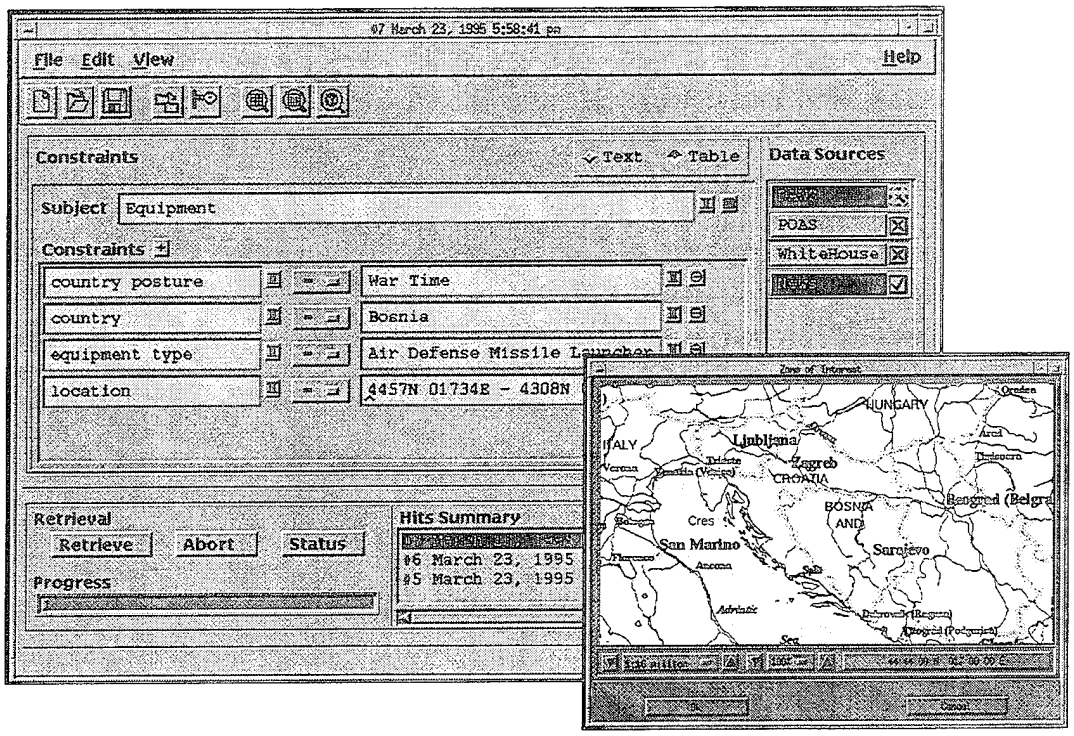


Figure 1: Users Can Geographically Constrain Queries via a Map Display

In addition, users may specify that they wish to retrieve data only from certain sources, or they may allow Envoy to determine the potentially relevant sources on its own. In practice, as a user becomes familiar with the available sources, he or she will tend to restrict the queries by source in order to minimize response time, but this capability is invaluable when dealing with new sources and new query subjects. Users are responsible for obtaining accounts on any sources they wish to access, and must supply Envoy with their account ID and password.

Finally, users may run multiple questions concurrently (although some sources allow only sequential execution of these multiple questions), and can save both questions and results for later perusal and potential reuses. In fact, a scheduling option currently under construction allows a user to automatically retrieve the latest information in his or her area of interest, so that the most current status is regularly available first thing in the morning when the user logs in.

3.2 Entity-Centric Presentation

While much of the business world rushes to embrace a document-centric view of the world, it is important to realize that such a view is not always appropriate for military matters. For example, while it is nice to be able to read about certain weapon system capabilities in document format, such an approach is not necessarily optimal for viewing the latest fighter positions in real-time, or for exploring the potential relationships between thousands of communication intercepts. Such a document-centric viewpoint also does not lend itself to reasoning about the returned information, and combining data from various sources to present a unified view of the situation.

Consequently, while Envoy naturally includes a default set of text, table, imagery, and video "document" viewers, it also provides seamless integration with a collection of entity-centric software applications that support further analysis and presentation of returned results. Rather than focusing on the document (i.e., formatting, page numbers, etc.), these entity-centric application focus on the individual real-world items the data represents, for example allowing users to plot positions of returned items on a map, or automatically adding units to a command hierarchy display. The focus on the *entity* enables users to work with ever-changing, dynamic data, in a sense creating a *document* on the fly that may combine data from several external data sources gathered through a series of related questions.

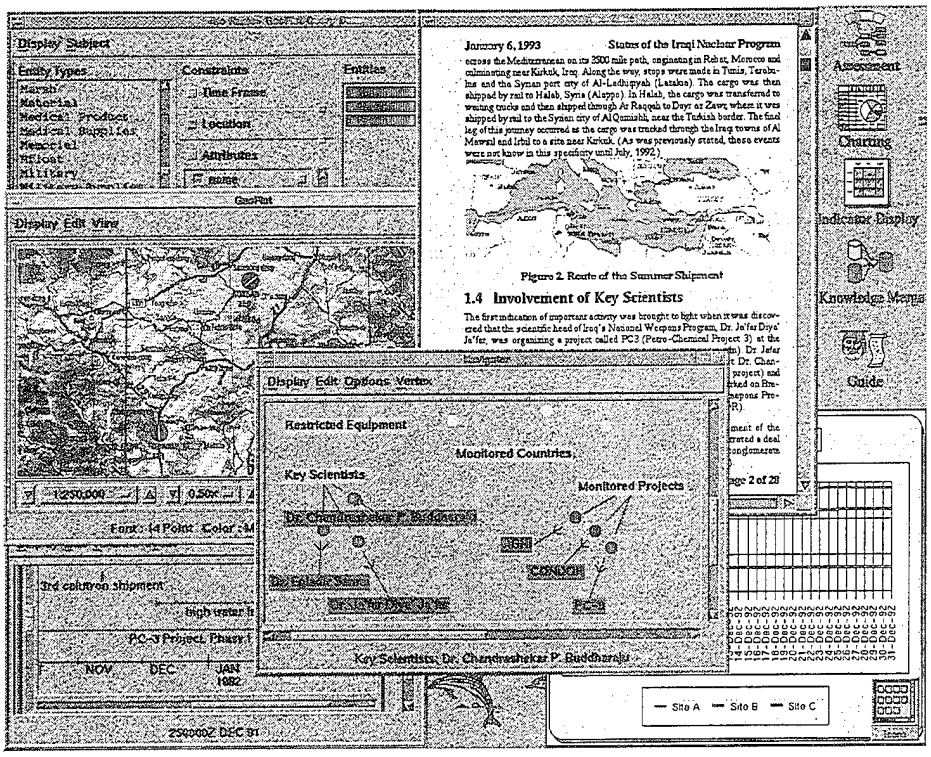


Figure 2: Extensive Options for Exploring Returned Information

More than just a simple conduit that retrieves specific data items from remote sources, Envoy provides a dynamic, object-oriented framework which allows users to express their questions and work with their results in terms of their individual problem domains. This framework can be revised as desired using built-in graphical editing tools, also enabling users to quickly accommodate changes to the types of information available, as well as to the problem domain itself.

3.3 Connection to New Sources

Envoy contains three basic capabilities to assist with the development and maintenance of data source connections. The *Source Explorer* automatically reads the structures of external databases and presents them to the administrator in preparation for mapping the entity-centric knowledge framework to the structural organization of the remote source. The Source Explorer also flags changes to the external source, facilitating maintenance of existing Envoy connections. The *Concept/Source Connection Facility* enables administrators to specify the mapping between Envoy concepts and the data structures returned by the Source Explorer. For example, administrators use this facility to indicate which table columns should be used to populate concept attributes, as well as more complex issues such as navigating the external source structures to find related information, and selecting the default types of information to return (unless overridden at execution time by the power user). Finally, the *Distribution Facility* enables the

administrator to automatically update all user knowledge bases on the network to begin using new connection information.

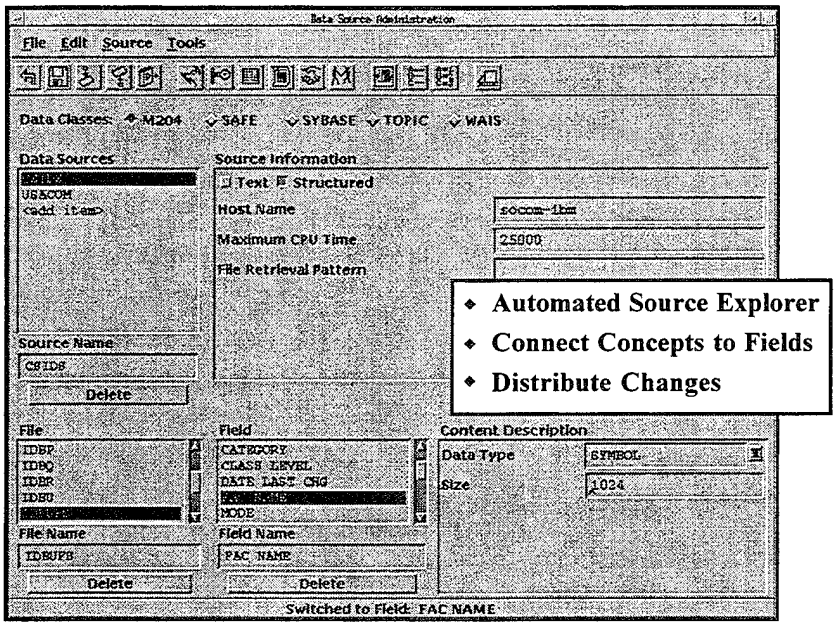


Figure 3: Envoy Administration Includes Automatic Source Exploration Capability

In the future, the DOD will utilize only a select few "migration" systems, drastically simplifying the data access environment. Until then, local system administrators will be faced with the burden of providing their users access to a wide spectrum of local and remote data sources which are not already prepackaged within the delivered Envoy system (actually, this burden on local administrators may continue far longer than expected, due to local desires for different knowledge frameworks...i.e., some organizations will want different types of information extracted from the same sources in different frameworks...and also due to the proliferation of commercially available data sources which the DOD does not manage and which will provide an increasingly rich source of information about a variety of subjects).

3.4 Envoy & Intelink

The DOD has selected a combination of Internet tools (i.e., Mosaic, WAIS, etc.) as its preferred method for data dissemination, calling its implementation of this architecture "Intelink". Intelink tools are remarkably easy to use, have the capability to support excellent data presentation, and at this point, are devoid of software costs. Intelink represents a substantial leap forward in data dissemination capability.

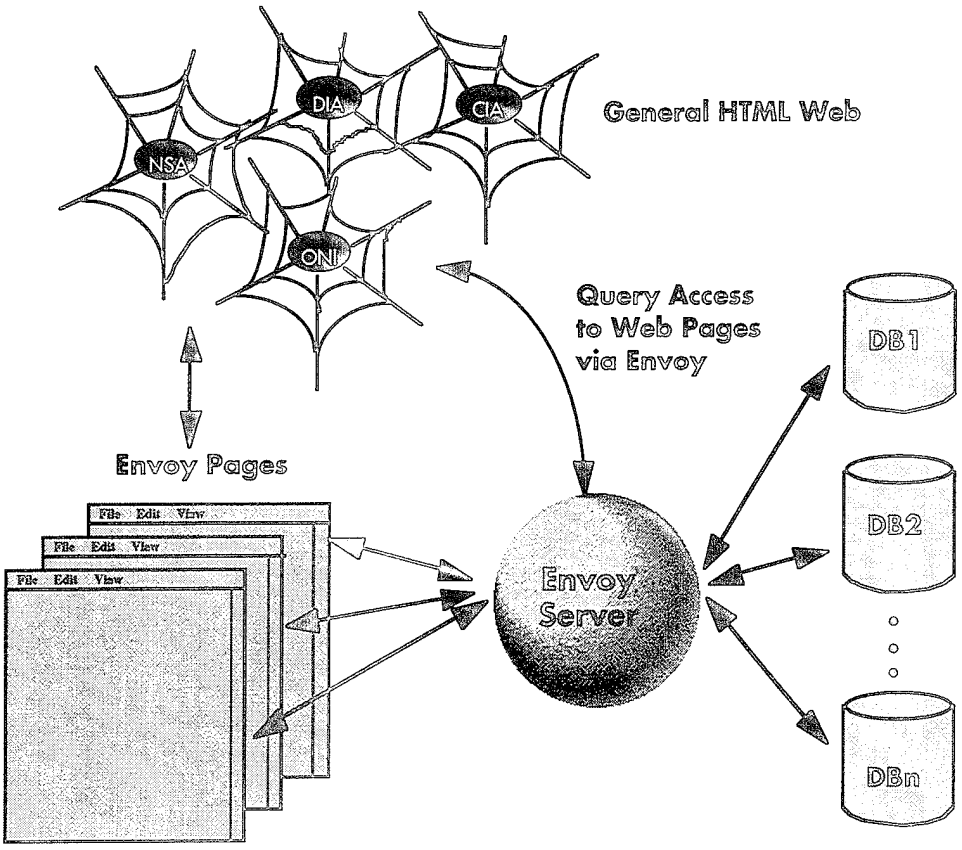


Figure 4: Intelink Documents Will Soon Be Accessible Through Envoy

However, Intelink currently does not provide a mechanism for accessing legacy systems or even standard SQL databases such as Sybase or Oracle. Rather than trying to force all DOD data into document format, ARPA is funding an extension to Envoy that will allow typical Intelink users to retrieve data from these structured sources, all from within standard Intelink HTML documents and forms. Users requesting data through these Intelink forms will launch standard HTTP requests that the Envoy server satisfies, returning the results of its queries in a dynamically created HTML document (albeit with minimal formatting). In addition, Envoy will also soon add Intelink as one of its supported data sources, so that submitted questions may search portions of the Intelink web in addition to any other applicable sources, such as Topic message repositories or Oracle databases.

Although we expect the majority of DOD users to utilize Intelink as their primary data access tool, thereby interacting with Envoy only as a server, we nevertheless expect numerous analysts to continue using the Envoy user interface directly, as it provides more expressive query formulation (for example, drawing geographic restrictions on a scaleable, adjustable map) as well as more powerful, entity-centric tools for manipulation and presentation of results, as described in the preceding section.

4. ENVOY ARCHITECTURE

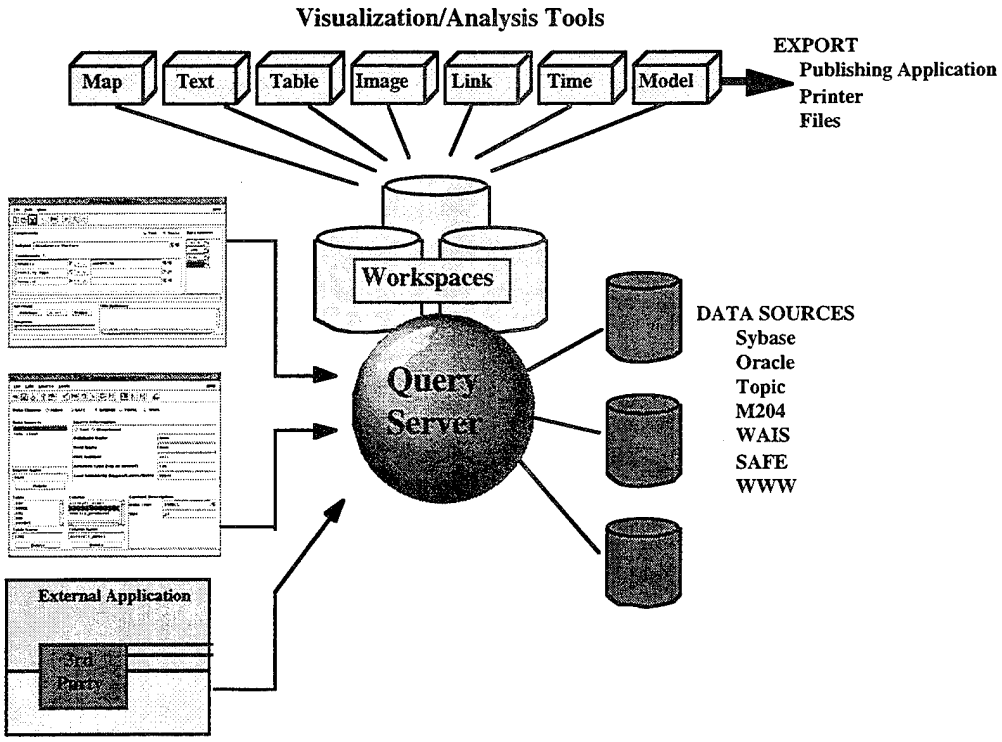


Figure 5: Modular Architecture Enables Envoy to Serve Multiple Front Ends

The Envoy architecture divides external access into three major components: question interfaces, query server, and visualization and analysis tools. The components are distributed and act independently using multiple workspaces to coordinate their tasks. Communication channels are optimized between each component. For example, the Query Server communicates with data sources using the native language of the data source and the visualization tools use an object-oriented information access library to retrieve query results from the workspaces.

4.1 Question Interfaces

Question interfaces allow a user to query external databases using their own concepts and terminology. A *Question* is made up of concepts. Concepts are related to physical database constructs using an administration interface which supports complex joins, nested queries, and code translations. Concepts have attributes which may be constrained and/or returned. Potential constraint values are associated with attributes so that the interface may provide them upon request.

The resultant *Question* construct is sent to a Query Server for processing. The Envoy-supplied X-Windows question interface communicates with the Query Server using

Tooltalk, a standard interprocess communication mechanism; other front ends can communicate with a Query Server using either CORBA, HTTP, or of course, Tooltalk.

The Query Server supplies information to the question interface in order to allow it to report the progress of the resulting queries, including number of hits, connection errors, or other query related warnings and errors. Aborting a question is initiated by the question interface which sends a message to the Query Server to stop the given question and report back any results obtained up to that point.

Questions, and their associated results, may be saved for future examination. In addition, saved questions may be recalled and sent to the Query Server with little or no modification. Notes may be attached to saved questions.

4.2 Client/Server Query Management

The Query Server is an independent process that receives questions, translates them into native queries, sends the queries, and then processes the results. The Query Server generates query sub-processes to service a question. The number of sub-processes is controlled by the Query Server to make optimal use of the workstation memory and processing resources.

The Query Server is composed of two modules: connection and session. The connection module handles the native interface to the databases. The primary functions are open, close, test, query and abort. The session module handles the translation of questions into queries and the whole query process from opening the connection to processing the results. Connection modules may be shared between two similar databases (e.g. legacy systems using 3270 terminal emulation). In the same way, session modules may be shared (e.g. Sybase and Oracle both use SQL).

4.3 Visualization Tools

Results from queries are stored in a common workspace that can be accessed with a variety of tools from text and table viewers to link analysis and timeline tools. The tightly integrated visualization tools provide a mechanism to export the query results to other media such as printers or files and to publishing or other presentation applications.

5. DEPLOYMENT

Envoy has been operationally deployed, at least in prototype form, since early 1993, and is currently in operation at Atlantic Intelligence Command (AIC), U.S. Special Operations Command (USSOCOM), III Corps, I Corps, 18th Airborne, and other Government agencies. A commercial version of this software has been created by Delfin Systems, and has also been deployed at various domestic and international customer sites. Although

there are several commercial components embedded in the Envoy software, for a limited time interested parties may contact CDR Tom Cool at 301-669-5228 to obtain copies of this software at no cost.

Interestingly, our deployment experiences have demonstrated that a small fraction of those who could benefit from the data already available in multiple data sources actually know how to access this data and utilize it. In one case, we were unable to find a single person in an entire organization who knew how to run a query using the standard interface for an important, widespread database, so that we could verify that Envoy was retrieving the correct data. In another case, we received complaints from a database administration staff because too many queries were being run against the database, considerably slowing overall performance. Upon investigation, we discovered that two or three simultaneous users — on this mainframe — could cause serious performance problems. This limitation reveals the extent of the utilization this database received until Envoy arrived to make it available to more than a core set of technological elite.

To be fair, many analysts believe that the data contained in these databases is hopelessly inaccurate and outdated, even when an ostensible effort is put forth to maintain the information on a weekly basis. Given this attitude, many analysts question the effort to master the various arcane commands necessary to tease this suspect data from the various sources.

We expect that Envoy will help identify any such problems with existing and future data assets, so that gaps in knowledge can be clearly and quickly identified...and redressed. The fact is that Envoy not only simplifies data access...for many, it makes data access *possible* for the first time.

6. CONCLUSION

When the Envoy project began, there were few, if any, tools that could provide end-user access to multiple, heterogeneous data sources. Since that time, a number of heterogeneous access tools have appeared, some with impressive capability and robust performance. For the most part, however, these other tools tend to concentrate primarily on SQL databases, ignoring the wealth of information available in legacy systems as well as cutting-edge text repositories such as those found on Intelink. Moreover, these other tools lack the dynamic, object-oriented framework that makes Envoy so convenient for both administrators and end-users alike; they also lack Envoy's presentation options that are so well suited for entity-centric military computing. In any case, Envoy's modular architecture makes it quite straightforward to utilize these tools at the back end, when appropriate, to increase Envoy's breadth of connectivity, or even at the front end (as in the case of Intelink), to add power to a narrowly focused application.

It is our vision to provide a standard commodity for heterogeneous data source access within the DOD, and with the initial version complete and beginning distribution throughout the community, we are well on our way. Naturally, we can see many areas for improvement within the application, and we hope that combined interest from both the Government and the commercial world will enable us to see this vision to fruition. For now, our immediate future is focused on additional Envoy deployments, user support, and integration with Intelink. We highly encourage any interested organizations to contact us as listed in the *Authors* section.

7. AUTHORS

All authors work for Delfin Systems, and can be reached by telephone at (408) 748-1200, or by fax at (408) 748-1140. Email addresses are given for each individual author, and U.S. mail can be sent to: Delfin Systems, 3000 Patrick Henry Drive, Santa Clara, CA 95054.

Mr. Dan Stickel (dan@delfin.com) is currently Vice President and General Manager of the Decision Systems Division at Delfin Systems. He has been the Program Manager of the Envoy program since its inception, and is also responsible for Delfin's InfoPower™ line of information analysis software. Prior to joining Delfin in 1987, Mr. Stickel worked at AT&T Bell Laboratories, where he received an award for "Extraordinary Contribution" in the field of international planning tools. Mr. Stickel holds a BA and an MS in Computer Science from Harvard University.

Mr. Tom Hillman (hillman@delfin.com) is currently the Software Development Manager with Delfin's Decision Systems Division. He has been the Envoy Project System Engineer since the program's inception, and is primarily responsible for the technical design of the software. Mr. Hillman is also involved with a variety of other software technologies, including associate systems and expert systems. Prior to joining Delfin in 1989, Mr. Hillman worked for the Knowledge-Based Systems Group at Michigan State University. Mr. Hillman holds an MS in Computer Science from Michigan State University.

Mr. Larry Safran (larry@delfin.com) is currently a Staff Engineer with Delfin's Decision Systems Division, and has worked extensively on a variety of aspects of the Envoy software, with particular emphasis on data source connection administration. Mr. Safran also has primary responsibility for redesigning InfoPower™'s underlying data representation to increase performance. Prior to joining Delfin in 1994, Mr. Safran worked at IBM for 9 years, where he developed a variety of administration tools, integrated environments and earned an "Outstanding Technical Achievement" award. Mr. Safran holds an MS in Computer Science with an emphasis on database technology from Stanford University.

Mr. Jerry Beersdorf (jerryb@delfin.com) is currently Program Manager for the Intelligence and Objects Data Base Generator (I&O DBG) extensions to Envoy for the WARBREAKER program. He has worked extensively on many aspects of the Envoy software, with the primary focus being on native M204 code generation. Mr. Beersdorf has extensive experience with the development of data base applications for SQL based systems such as Sybase and Oracle. Prior to joining Delfin in 1986, Mr. Beersdorf served in the United States Navy at shore and afloat commands for 20 years as a Cryptologist. Mr. Beersdorf holds a B.S. in Computer Science from North Carolina State University and an M.S. in Computer systems Management from the Naval Postgraduate School.

A Technology Survey of Heterogeneous Data Access Across Multiple Data Types

Dr. David D. Mattox
Patricia Carbone
Dr. Marcia Kerchner
Ruth Hildenberger

The MITRE Corporation

Abstract. As part of the Access, Retrieval, and Information Exploration Study (ARIES) for the Federal Integrated Data Access Working Group (FIDAWG) and the Office of Research and Development (ORD), several leading COTS vendors and research organizations were visited to assess technologies that might be suitable for insertion into the analyst's workstation environment. Most COTS integration products provide a level of integration that solves the infrastructure heterogeneity problems (i.e., accessing multiple workstations holding a variety of structured DBMS products). Recently, some of the leading integration vendors have begun providing another level of integration by incorporating more than one data type into their products. Several consortiums and research organizations have addressed other Interoperability issues, such as semantic and representation heterogeneity, as well as how to accommodate complex data types such as imagery and geographic/spatial data in database systems. This paper will discuss near term trends in data integration products for structured data, text, imagery, and geographic/spatial data. Key words and phrases: data integration, information retrieval.

1. INTRODUCTION

Access to heterogeneous data types and systems is one of the most critical information management capabilities for the 1990's. Technology that supports such access could provide an environment for analysis that makes information gathering an efficient, powerful process. New activities at the national level, such as the proposed Information Highway, National Information Infrastructure (NII), and the High Performance Computing and High Speed Networking Applications Act, make the achievement of such a capability more imaginable than ever before. These programs have the potential to contribute to an environment where it will be possible for information retrieved from different databases to be at a user's fingertips.

This capability is especially critical today because not only has the amount of data to which most users have access increased dramatically (even in these pre-Information Highway days), but there is a need to access different sources of data, different database management systems (DBMSs), and different data types. It is no longer realistic to expect users to learn an array of interfaces to different DBMSs. It is equally clear that conceptual integration of this rapidly increasing flow of available data will be critical for users.

The Access, Retrieval & Information Exploration Study (ARIES) is part of the Federal Integrated Data Access Working Group (FIDAWG) program to improve methods for accessing and retrieving information distributed across heterogeneous systems. The study is currently in progress and is looking at the state of the research directed to solving aspects of the heterogeneous data access problem as well as the state of the commercially available technology in order to gain insight into existing and evolving technologies. The final report will contain a strategic technology insertion and research plan whose purpose is to be a guide for technology insertion opportunities and to identify areas in which to promote research and development of certain technologies. These technologies can help achieve Agency goals for accessing and retrieving four types of data: structured (i.e., highly formatted) data, free form or semi-structured text, geographic/spatial data, and imagery. This paper is a synthesis of a previous version of the study [1] and the current ARIES effort.

With the successful development of technology to store large amounts of data over the past 15-20 years, many systems have been developed to acquire and manage that data for a variety of applications in all parts of the government, industry, and academia. Prior to seven years ago, the systems were primarily built on large mainframe computers. The applications tended to be large stovepipe systems that did not allow the sharing of resources with other applications. More recently, capabilities of individual workstations have increased, so many newer applications are beginning to take advantage of the ability to share data and other resources across networks.

The problem is that many of the older applications and data sets still cannot be shared easily, since each was developed as a self-contained system. The desire and need to share data across applications and to provide an integrated or corporate view of the data for analysis have been frustrated by the inability to cope with the issues of heterogeneity among the systems and applications. A measure of the increased interest in this problem is the number of meetings within the past five years dealing with the issue of how to integrate these databases and systems, including workshops on Semantic Heterogeneity and Interoperation in Multidatabase Systems in 1989, 1990, and 1992 [2], a workshop on Heterogeneous Database Systems [3], international workshops on Interoperability in Multidatabase Systems in 1991 and 1993 [4] [5], and workshops on the Intelligent Integration of Information in 1993 [6], the AAAI Symposium on Information Gathering from Heterogeneous, Distributed Environments [7], to name a few. In addition, several technical journals have devoted whole issues to special reports on integrating systems [8] [9]. Together with President Clinton's goal of providing Information Highways and the need to reduce the cost of storing the same data multiple times, the motivation to provide intelligent access to different data management systems is clear.

Heterogeneity has many aspects. It includes differences between the following:

- Data management systems or tools;
- conceptual data models (e.g., relational, object-oriented, or hierarchical);
- semantic or conceptual data schemas (meanings and relationships of the data);
- database content;
- data type (e.g., imagery, geographic, video, text, or structured);
- computer systems (platforms);
- transaction management procedures;
- application-specific abstractions; and
- data format.

Research and tools to address the goal of providing heterogeneous data access must consider these various types of heterogeneity, both as separate issues and as a whole. The technical assessment in this paper focuses on the first five differences, noting that differences in data type lead not only to data access but to data display issues.

The current commercial technology for heterogeneous data access focuses on merging schemas of multiple relational databases, some file systems, and a few of the more popular pre-relational DBMSs, such as IMS or IDMS. Many of the current COTS products allowing heterogeneous access have concentrated on solving the infrastructure heterogeneity problems between computer systems, such as differences between workstation architectures and differences between network carriers. Much of the current research concentrates on other heterogeneity aspects, such as the various forms of semantic heterogeneity and representation heterogeneity. The current trends in technology for the four data types (with an emphasis on access between systems that handle them) as well as more general developments in data integration are discussed in the remainder of this paper.

2. STRUCTURED DATA TRENDS

There are four main types of structured databases available commercially, file-based (e.g. IMS) relational (RDBMS), object-oriented (ODBMS) and extended-relational (ERDBMS). Of these, this paper will concentrate on the last three. This section addresses two main issues important to structured database systems, access to multiple systems and storage of multiple types.

Structured database vendors and researchers are attacking the data access issues in two different ways. One method is through the use of gateways to other databases, and the other method is by incorporating multiple data types in a single database. Over the past ten years, many database vendors have developed distributed versions of their products that handle transaction management and data access quite well. Currently, most database vendors have begun building gateways from their own products to other structured DBMS products as well as file systems. The database gateways provide client applications with an Application Programming Interface (API) that lets various data sources or services appear equivalent. These gateway APIs have three different sources: established vendors such as Sybase or Oracle, standards bodies such as X/OPEN or the American National Standards Institute (ANSI), and industry consortiums such as the SQL Access Group (SAG). Presently, only the APIs of established vendors have garnered attention as gateways, largely because tools and applications are readily available for those interfaces.

The distributed gateway products (e.g., the Sybase / Microsoft SQL Server, Oracle's Open Gateway, UniSQL/M) handle differences between workstation architectures, network carriers, and structured DBMS products, and they handle transaction management to some degree. Primarily, queries are made using either the ANSI standard SQL-89 or SQL-92, although there are translations to native SQL queries. However, semantic heterogeneity issues between the database structures using the same data models or differences between relational and object-oriented data models are not yet solved, and much of the research is currently concentrating on these issues. ARPA's Intelligent Integration of Information (I^3) project is an effort to define and solve these problems.

Other than the gateways, structured DBMS vendors and researchers are attacking the problem of heterogeneous data access by expanding the capabilities of the current database systems to store different types of data, for example, through the Binary Large Object (BLOB) data type for storing images or long text strings. As well, RDBMS companies are adding new data types and associated access and indexing methods to better manage specific, domain-dependent problems. The object-oriented data model has had a significant effect on the structured DBMS market, as numerous object-oriented database companies, such as Object Design Inc.'s ObjectStore, and Objectivity Inc.'s Objectivity/DB, are now competing with the well-established relational database vendors. RDBMS vendors such as Ingress, Oracle, and Sybase in turn are expanding their products to allow the description and storage of more complex data structures or objects. Oracle 7 MultiDimension provides a good example of this by defining a new data type and extended SQL functionality to manage spatial data. Illustra also provides many new data types and access methods through the use of its DataBlades™ add on software.

Several of the current structured database vendor and research products allow text, audio, geographic/spatial, and imagery data to be stored in the same database in addition to the traditional structured data through the use of BLOBs or new data types. A few structured database vendors are combining object-oriented technology with relational technology into hybrid systems, such as Illustra System's Illustra, Hewlett-Packard's OpenODB, and UniSQL Corporation's UniSQL, because they want both the powerful capabilities of an RDBMS (including the relational query language, SQL) and the flexibility of objects, which can be data of any type. Additionally, there is progress being made in defining the next SQL standard, SQL3, which is due to be finalized in 1996 or 1997. The SQL3 standard [10,11] will be an extension of the current SQL standard and systems adhering to it will provide basic object-oriented capabilities and many new data types. [12]

3. FREE-TEXT TRENDS

It has been estimated that approximately 200 gigabytes of new textual data available to the intelligence community are added to on-line archives each year [13]. Not only does the amount of text present problems, but there is a high level of heterogeneity because of the variety of sources, particularly for open source materials. The three main standards for information retrieval are the Gopher protocol, the World Wide Web (WWW), and the Z39.50 protocol. The open source community is beginning to converge on the Z39.50 protocol as a standard for communicating with text databases because of the popularity of the Wide Area Information Service (WAIS), which provides access to over 450 databases and is based on that protocol. Although no one standard suffices for all aspects of an integrated network, the acceptance of such a standard will certainly facilitate access to many sources of text. However, the query capabilities available in many of these systems are keyword or statistical-based and limited in their accuracy (accuracy usually being measured as precision, the percentage of retrieved material that is relevant, and recall, the percentage of relevant material that is retrieved). These systems would certainly benefit from some of the new, promising text retrieval approaches, such as those being developed in the ORD and Advanced Research Projects Agency (ARPA) sponsored Tipster projects.

Modern text processing products today have some collection of the following features: relevance ranking, simple natural language queries, query by example, term weighting,

Boolean retrieval, simple database attributes, query formulation assistance (such as thesauri), a variety of interfaces, APIs, and a client-server environment. The first four features are those that users expect today. There are currently two categories of text retrieval models on which products are based: traditional, e.g., exact match Boolean combinations of index terms, statistical vector space model, extended Boolean (allows "fuzzy" retrieval), and probabilistic (estimates the relevance of a document based on statistical properties of word frequencies), and the non-traditional, e.g., ruled-based (or inference-based) and neural net-based. Most of the commercial products are based on traditional models with a few newer systems emerging that use one of the non-traditional models. To deal with data display, there is a trend in commercial products toward text retrieval engines, such as Fulcrum's Ful/Text, rather than complete packages because user interfaces are so application-specific. Most of the commercial systems provide application development environments (sometimes through third-party partnerships) for development of user-specific front-ends, rather than or in addition to generic ones. Thus different analytic applications could access the same databases but use different front-ends for that access. In addition, providing text retrieval as an engine has led to commercial efforts to incorporate such capabilities into a structured database environment through the use of APIs.

Gopher is an example of the client-server tool approach that gives menu-based access to documents and directories across the Internet. There is a new version of Gopher, called GopherPlus, that uses meta-information and template forms and will have gateways to Sybase and Oracle servers. The directory services available on the Internet have several shortcomings. Archie returns too many hits to search through, Veronica takes two days to build an index, the WAIS Index of Servers relies on the descriptions that are submitted to think.com, and X.500 is non-searchable (although there is a project called Nomenclator at the University of Wisconsin that is making the X.500 directory service searchable) [14].

Another creative approach to user interface technology supporting the display of large information (particularly document) spaces is Xerox Palo Alto Research Center's (PARC's) Information Visualizer, parts of which are currently available commercially from XSoft [15, 16]. Its Cone Trees feature can display a large amount of data at once on a 3-D variant of a pyramid chart, on one screen, with an upright cone tree or a horizontally-positioned one. Labels hang on the image, each representing a set of data that can be selected, moved, enlarged, or linked with other data sets. A cone tree can be spun to bring a particular label to the fore. The Information Visualizer uses agents for conducting searches, organizing information into clusters (using the "scatter/gather" algorithm [17]), or designing presentations of information. The search agents use keyword or relevance feedback search mechanisms to find documents. To be discussed later is the potential use of the Information Visualizer to support a data directory for multiple databases. There is also ongoing research focusing on the refinement of clustering techniques as well as COTS products that perform some similarity clustering to aid in searching and displaying document spaces. An approach to providing a representation that is a visualization tool as well as a visual query language is the Massachusetts Institute of Technology's (MIT's) InfoCrystal [18], which transforms relationships among concepts in a query into an iconic representation.

4. GEOGRAPHIC/SPATIAL TRENDS

Geographic Information Systems (GISs) and image processing systems have been converging over the last few years to the point where today much functional overlap exists between them. With this convergence comes the development of a broader data categorization called *geospatial information;* it includes geographic, cartographic, and imagery data as well as any data type with a geolocational component. In this section, the family of geospatial software products included are general purpose automated mapping systems and GISs. In the next section, Imagery Trends, image processing software products and research trends are described.

An automated mapping system is designed to perform basic mapping operations, such as displaying maps in various scales and projections, calculating geographic coordinates for given points, or calculating distance, areas, bearings. Most mapping systems essentially provide playbacks of digitized paper maps with the ability for the user to overlay specific points and features. Some mapping systems can also generate various three-dimensional displays such as line-of-sight (LOS) or perspective views, zoom in or out, or generate overlays to display additional information.

In contrast to mapping systems, GISs are comprehensive information systems designed to capture, store, process, integrate, analyze, and present geographic/cartographic data. This technology embraces the disciplines of cartography, photogrammetry, visualization, database management, image processing, and communications. Since their inception in the early 1960's, GISs have helped automate many geographic/cartographic operations (e.g., feature overlay and terrain analysis) that were previously performed manually with great difficulty and expense. More recently, GIS technology has made possible a whole new spectrum of processes such as spatial modeling and the integration of disparate spatial data sources, including imagery.

Whereas mapping systems may be most appropriate for applications requiring background situation displays, GISs provide the flexibility to accommodate a variety of applications and integrate a variety of data types. Sophisticated data manipulation techniques and analysis functions such as image processing, spectral analysis, water drainage path predictions, and optimum path calculations may also be included in a GIS package.

Limitations of today's technology are that most GISs are complex and difficult to learn, are not based on standards, and are slow to transition to an open systems environment. Their APIs are still immature, and their user interfaces are not intuitive. In addition, full availability of digital data is still on the horizon. Specific development activities and directions in which geographic/cartographic information systems are heading are:

- The use of object management technology to provide interoperability among heterogeneous systems in the geospatial information community;
- The standardization of APIs to support the use of GIS and mapping capabilities as embedded applications in larger systems;
- The addition of geographic/spatial data types in structured DBMSs;
- The development of better means to assess and report the accuracy and lineage of spatial data especially when combining data from different sources;

- The integration of spatio-temporal data concepts in spatial data models; and
- higher-level, more intuitive consistent user interfaces.
- The integration of geographic/cartographic information systems and image processing systems.

The Open GIS Foundation (OGF) is developing API standards for the GIS community with its Open Geodata Interoperability Specification (OGIS). OGF is a consortium of private, public, and academic organizations with experienced backgrounds working towards non-proprietary geospatial data interoperability methods. Recently held OGF meetings included executive briefings designed to increase the awareness of activities involving spatial data models, object management architectures, and multi-application information systems, and to obtain input from those involved in geodata standards activities.

5. IMAGERY TRENDS

Until the mid-1980's, digital image processing had been a narrow discipline that was utilized almost exclusively by government and academic organizations. As with GISs, the development of powerful computer workstations and personal computers has made image processing available to a much broader user base. This development has resulted in a proliferation of image processing products with varying capabilities, from very specialized to general purpose. Although a few image processing systems have recently added heterogeneous data access to vector map (spatial) data, no commercial image processing software products provide heterogeneous data access to structured data or free text. This section describes the COTS products and research trends in the image processing industry.

Image processing products available today may be commercial, GOTS, or public domain packages. Imagery analysts in the intelligence community use "high-end" photogrammetric workstations with specialized hardware components for the computer-intensive algorithms associated with exploitation of images and sophisticated display monitors and printers. These systems are customized by vendors for image exploitation. A "mid-range" package is one that provides a broad range of functions, including, but not limited to: complete geometric transformation capabilities; multi-spectral processing; a comprehensive suite of image filtering, intensity, and detection algorithms; and support for graphical annotation and symbology. Mid-range packages (ERDAS IMAGINE, PCI EASI/PACE, and Terra-Mar's IDIMS) have the advantage of being comprehensive and providing virtually any function for the user. The potential disadvantages include possible lack of software modularity (for embedding applications), the need for large amounts of disk storage (for executable code), and complexity of use in native mode.

A "low-end" package has a subset of functional capabilities found in the high-end packages. Low-end packages are often targeted at users with special requirements (e.g., intelligence analysts whose primary task is not imagery analysis). The functions available in these types of packages usually will satisfy image processing requirements for such "niche" users. In general, the advantages of using a low-end package include ease of use, relatively small storage requirements for executable code, and comparably low cost. The disadvantages may include limited functions and inadequate documentation. It should be

noted that packages within these categories do vary in the particular functions that they support and in the quality and comprehensiveness of how they are implemented.

Many of the vendors of low-end packages are adding functions to their products that have been traditionally available only in mid-range and high-end products. Also, as image processing systems (and GISs) continue to evolve, they are increasingly including more general *spatial information processing* capabilities in single packages. For example, some image processing packages are starting to support the integration of digital cartographic data with imagery.

A potential approach for image data management may be through the use of an OODBMS. An OODBMS provides the same capabilities as an DBMS, and it also has other features that are well suited for managing complex data sets such as images. Complex combinations of images and other information (e.g., image products) can be more easily stored, processed, and retrieved in the object model than in the relational model. However, the emerging SQL3 standard addresses the storage of images in an RDBMS beyond the BLOBs in use today. As well, the Extended-Relational DBMS Illustra provides an add-on capability to manage images using its Image DataBlade which allows the retrieval and manipulation of images. IBM also has Query By Image Content (QBIC), a front-end to a database which allows the user to retrieve images based on their characteristics and content. Other research and development activities include the development of interactive information systems to browse catalogs and directories of image data using a geographical user interface (an index map of the world) and make requests for specific images. Also, the RADIUS project is performing research to validate the concept of Model Supported Exploitation (MSE), as well as conducting a review and tradeoff analysis between existing Image Understanding (IU) technology (including registration of images to points on the earth) and imagery analysts requirements.

6. DATA INTEGRATION TRENDS

At present, some intelligence analysts can have up to four terminals on their desks to access various databases. Ideally, analysts should be able to deal with multiple databases using one language, independent of the underlying DBMSs, operating systems, networks, or terminals. According to a group of researchers gathered for a conference on interoperation in multidatabase systems [2], "the basic conditions for *interoperability* between heterogeneous computing systems (file transfer, remote login, etc.) either have been achieved or will be achieved in the near future." It was agreed, however, that "this system-level interoperability is not by itself sufficient to allow development of complex applications spanning the boundaries of multiple systems that were designed to operate in a stand-alone mode." It is clear that the following heterogeneities must still be resolved:

- Differences among the data models;
- differences among meanings of and relationships between the data in one database versus another (which cause semantic heterogeneities);
- differences among data formats (causing representation heterogeneities);
- potential overlaps and dissimilarities in database content (causing population heterogeneities);
- disparate data types; and
- differences among transaction management strategies.

There are several different "integration" scenarios that may be appropriate for different applications or uses. One involves the development of a global schema from all the component database schemas, thereby requiring that all the schema heterogeneities be resolved, new views of the integrated data be developed, and transaction management be handled consistently across the integrated system. In this case, the component databases become subservient to the integrated system. The Microelectronics and Computer Technology Corporation's (MCC) Carnot project (now called InfoSluth) handles this type of schema integration through the mapping of the individual database schemas to the Cyc knowledge base, while Data Integration's InterViso provides a data dictionary tool to handle data structure and content incompatibilities when the integrator creates the global schema. There is one research project underway at Northwestern University called SEMINT which addresses the issue of automatically generating metadata for legacy databases using neural nets to analyze the content of the databases. [19]

A somewhat lower level of integration is a federation of databases. Each component database defines what data is available for the integrated view (their export schema), and these export schemas are integrated into the federated view that users can access. However, the component databases maintain their autonomy. Several data integration products such as InterViso, Uniface Corporation's Uniface, and Information Builder's EDA/SQL can be used to create a system of federated databases, with some coordination between the component DBMSs and the data integration product. Unfortunately, there is no mechanism at the federation level for tracking changes to component schemas, so there must be cooperation between the component database administrators and the federation administrator, since changes to a component schema will affect the federation level schema.

The lowest level of integration involves the development of interoperable systems, where totally autonomous systems simply share data between them, with no need or use for any kind of integrated schema. The Stanford-International Business Machines (IBM) Manager of Multiple Information Systems (TSIMMIS) project and the Object Management Group (OMG) Common Object Request Broker Architecture (CORBA) effort are examples of efforts to create interoperable systems. Representation and semantic differences between the data are resolved through the addition of metadata and the use of mediators. At this time, some implementations of CORBA are available commercially.

In response to the needs to allow access and update across data management systems and to display the various data types with a single interface, commercial vendors and researchers have developed a number of tools. While, as described in Section 2, several vendors of structured DBMS products are creating gateways to other similar products, some are providing access via gateways from their own products to managers of other types of data, such as text database products (e.g., Total Recall to BRS/Search, Sybase and Oracle to TOPIC, Sybase to Fulcrum Ful/Text).

An alternate approach to the use of gateways is "middleware," which is software that acts as an intermediary among different data management systems and from data management systems to other applications. Data access in middleware products or database gateways is typically via SQL-89 or SQL-92, since most RDBMSs are compliant with one of those two standards. However, some middleware companies allow the use of "native SQL" queries, meaning that those queries would be passed directly to one specified database.

Note that this type of integration handles issues of heterogeneity dealing with the database infrastructure, but does not fully address the semantic and representation differences between the data. As with structured data, researchers looking at heterogeneous data access are studying how to deal with those issues of semantic and representation heterogeneity. It is hoped that the outcome of this research and development will result in the development of new standards to ease the development of distributed, heterogeneous systems.

An object-oriented paradigm that could have a profound effect on the development of data integration capabilities is distributed object management (DOM), in which all entities are treated as objects. Clients access objects using a single, uniform programming interface, without regard to where and how the objects are implemented [20]. HyperDesk Corporation and Digital Equipment Corporation (DEC) have both developed computing environments based on DOM. The DOM works using the concept of the Object Request Broker (ORB), which, as defined by the OMG, provides the mechanisms by which objects transparently make requests and receive responses. The various types of data (and other applications) are treated as defined objects. Applications can then access these defined objects and use them according to their definitions. The ORB provides interoperability between applications on different machines in heterogeneous distributed environments [21]. CORBA defines a framework for different ORB implementations to provide common ORB services and interfaces to support portable clients and implementations of objects (Figure 2). Instead of having to write an interface between each pair of objects (n*n interfaces), each object needs an interface only to the ORB (n interfaces).

ARPA is promoting research in the area of heterogeneous data access through its Intelligent Integration of Information (I3) program. The purpose of the I3 program is "to establish a technology and support the science needed to present information in a form and at a level of abstraction needed for high-level applications." [22, 23] The program will address such issues as metadata management, merging of data, and summarizing of data, all through intelligent agents that use knowledge of the user. The program is motivated by the idea that "automation of intelligent integration of information systems will relieve the user of tedium and permit better performance at less effort."

ACKNOWLEDGMENTS

We would like to acknowledge Ms. Jane Harmon, our COTR at ORD. We would also like to acknowledge Mr. Michael Josephs, the overall ARIES Project Leader. In addition, the following people participated in the technology visits or provided feedback on the Strategic Research Plan: Mr. Stephen Hirsch, Mr. William Jameson, Ms. Adrienne Kleiboemer, Dr. Len Seligman, and Mr. Michael Zoracki.

REFERENCES

[1] Kerchner, M. D., et al. Access, Retrieval and Information Exploration Study (ARIES) Strategic Technology Insertion and Research Plan. MTR 94W0000005, The MITRE Corporation, November 1993.

[2] Drew, P. , et al. Report of the Workshop on Semantic Heterogeneity and Interoperability in Multidatabase Systems. University of Houston Technical Report #UH-CS-92-13, Houston, TX, August 1992.

[3] Scheuermann, P., et al. Report on the Workshop on Heterogeneous Database Systems. SIGMOD Record, Vol. 19, No. 4, December 1990.

[4] Proceedings of the First International Workshop on Interoperability in Multidatabase Systems. IEEE CS Press, Kyoto, Japan, April 1991.

[5] Proceedings of the Second International Workshop on Interoperability in Multidatabase Systems. IEEE CS Press, Vienna, Austria, April 1993.

[6] Systems Workshop, Proceedings, ARPA/ORD-Sponsored Workshop on Intelligent Information Integration. Reston, VA, March 1993.

[7] Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments. Stanford University, March 27-29, 1995

[8] Special Issue: Semantic Issues in Multidatabase Systems. SIGMOD Record, Vol. 20, No. 4, December 1991.

[9] Special Issue: Heterogeneous Distributed Database Systems. Computer, Vol. 24, No. 12, December 1991.

[10] Melton, Jim (Ed.) ANSI Document number X3H2-94-329 and ISO document number DBL:R10-004 Working Draft Database Language SQL3, Aug. 1994.

[11] Melton, Jim Object Technology and SQL: Adding Objects to a Relational Language. IEEE Data Engineering, Vol. 17, No. 4, December 1994.

[12] Celko, Joe. SQL in the City of Steel: (The 1994 SQL Standards Committee Meeting in Pittsburgh PA). DBMS, Vol. 7, No. 4, December 1994.

[13] Lavender, B., et al. Massive Digital Data Systems for the Intelligence Community. Draft, The MITRE Corporation, September 1993.

[14] Reed, T. Interop '93 Trip Report. Electronic memo, The MITRE Corporation, October 20, 1993.

[15] Robertson, G., et al. Information Visualization Using 3D Interactive Animation. Communications of the ACM, Vol. 36, No. 4, April 1993.

[16] Information Technology Special Report. Fortune, Vol. 128, No. 7, Autumn 1993.

[17] Cutting, D. R., et al. A Cluster-Based Approach to Browsing large document collections. Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. June 1992.

[18] Spoerri, A. InfoCrystal: A Visual Tool for Information Retrieval & Management. Proceedings, Second International Conference on Information and Knowledge Management, Crystal City, VA, November 1993.

[19] Wen-Syan Li Knowledge Gathering and Matching in Heterogeneous Databases. Proceedings of the AAAI-95 Spring Symposium Series on Information Gathering for Distributed, Heterogeneous Environments, March 1995

[20] Proceedings of the Fifth International Workshop on Research Issues on Data Engineering: Distributed Object Management. Taipei, Taiwan, March 1995

[21] Object Management Group. Object Management Architecture Guide. Revision 1.0, MG TC Document 90.9.1, 1990.

[22] Winslett, M. The Winds of Change,. SIGMOD Record, Vol. 21, No. 4, December 1992.

[23] http://hafia.isx.com/pub/I3

AUTHORS

Dr. David D. Mattox is a Senior Member of the Technical Staff in Advanced Information Technologies Department at the MITRE Corporation in McLean VA and is involved in a number of tasks in the areas of advanced database systems and artificial intelligence. He can be reached at The MITRE Corp. 7525 Colshire Dr., McLean VA, 22102. Voice: (703) 883-6618. Fax: (703) 883-6435. Email: mattox@mitre.org

Patricia L. Carbone is a the Group Leader of the Intelligent Information Management and Exploitation group in the Advanced Information Technologies Department at MITRE.

Dr. Marcia D. Kerchner serves as an Associate Department Head of MITRE's Digital Libraries Department, directing work to assess and develop technologies supporting the management of information in heterogeneous digital libraries, focusing on capabilities for information discovery, retrieval, and electronic document management.

Ruth A. Hildenberger is a Member of the Technical Staff in Department G055 at MITRE. She is involved in a number of tasks for the Defense Mapping Agency involving desktop mapping, geographic information systems, and spatial database development.

Dr. Marco Emrich
Senior Director, Advanced Technology Group
Cincom Systems, Inc.

"Options for Object-Oriented Persistence"

Biography

Dr. Marco Emrich is Senior Director of Cincom Systems' Advanced Technology Group with responsibility for the company's system software directions, strategies and marketing. In this role, Dr. Emrich is instrumental in setting the direction for TOTAL FrameWork™, the industry's first cross-functional business application assembling environment. His background includes management of the NAS Information Network Technology Group at Digital Equipment Corporation where he was responsible for the Distributed Database and Database Interoperability Technology areas.

Introduction

The past several years have witnessed the gestation period for a new generation of database technology. During this time, there has been a flurry of activity to develop and experiment with database systems that support an object-oriented data model or that extend the relational data model with some object-oriented facilities. These activities have been fueled by the emergence of a broad spectrum of database applications which relational database systems cannot support, as well as the increasing need to achieve another productivity leap in application development. As a result of these efforts, there is now a sufficient body of knowledge for the development of a commercially viable next-generation database system. This next generation of database systems, called object-relational DBMS, is an object-oriented-based architecture which combines the best capabilities of pre-relational, relational and object-oriented database technology into a single unified system. The object-relational system extends earlier data models with core concepts found in object-oriented programming languages, including encapsulation of data and programs, object identity, multiple inheritance, arbitrary data types and nested objects. As such, an object-relational database is the only viable alternative for intensive query processing on complex objects.

The object-relational model supports all features engineered into mature RDBMSs to support mission-critical applications. These include ANSI SQL data definition and query language, updatable views, automatic query optimization, access authorization, dynamic schema evolution, automatic concurrency control, automatic recovery from crashes (including media crashes), triggers, client/server architecture, and more. In actuality, this model goes well beyond most conventional RDBMSs in supporting such features as repeating groups, nested tables, multidimensional data, etc. But most significantly, it provides true object-oriented features such as arbitrary data types, methods, encapsulation, multi-level inheritance, and polymorphism.

Additionally, this model accommodates application access to heterogeneous pre-relational, relational and object-oriented databases using a single global view and single database language. Cincom Systems realized the importance of this revolutionary database management sytem technology and incorporated it into TOTAL FrameWork—the industry's first object-oriented

application assembling environment for the development of cross-functional business applications. Using the powerful TOTAL FrameWork technology, organizations will no longer develop applications, they will assemble them. Creating and reusing objects, they will be able to deliver extensible, scalable and maintainable enterprise-wide applications.

Object-Relational Database Management Technology

In order to better understand the reason for selecting an object-relational database management system as the repository for objects in the TOTAL FrameWork, it is helpful to briefly review the evolution of database management systems from file systems through hierarchical, network, relational to most recently object-oriented database management systems. As systems have evolved, the data management activities, such as concurrency control, authorization, etc., have been taken out of the developer's hands and put into the server itself. For example, while a file system forced the application programmer to implement concurrency by throwing locks in the application code, the first generation of database management systems (hierarchical, CODASYL, networked) provided automatic transaction management, automatic upgrading/downgrading of lock granularity for dynamic concurrency, powerful data navigation capabilities, etc.

In spite of the major benefits delivered by hierarchical, CODASYL and networked database management systems, the last decade watched a new generation of database management system —relational—dominate the database market for business data processing applications. The theory, implementation, and use of database systems became a major discipline of computer science. The simplicity of the relational data model, the dynamic management of a database, and the power of the SQL language for query processing have been accepted as vehicles for significant productivity enhancements in application development.

However, even as the acceptance of relational database systems spread, their limitations were exposed over the last decade by the emergence of various classes of new applications such as:

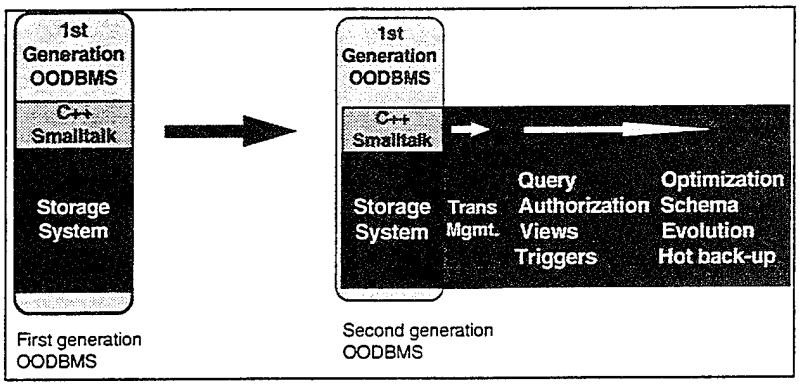
- multimedia systems
- statistical and scientific modeling and analysis systems
- geographic information systems
- engineering and design systems
- knowledge-based systems

The limitations of relational database systems exposed by these applications fall into two categories: data modeling and computational modeling. The data-modeling facilities that relational systems lack include those for specifying, querying and updating complex nested entities such as used in designing and engineering objects, and compound documents; arbitrary user-defined data types; frequently useful relationships, such as generalization and aggregation relationships; temporal evolution of data such as the temporal dimension of data, and versioning of data; and so on. The computation-modeling facilities that relational systems lack include the management of memory-resident objects for extensive pointer-chasing applications used for such things as the simulation of a computer-aided design; long-duration, cooperative transactions; and so on.

In order to solve the shortcomings of relational database systems, another fundamental advancement in database technology is required. The basis of this fundamental advancement is the object-oriented paradigm developed in object-oriented programming languages.

Solutions to most of the difficulties related to the data-modeling of relational database systems are inherent in an object-oriented data model. Relational systems are designed to manage only limited types of data, such as integer, floating-point number, string, Boolean, date, time, and money. In other words, they are not designed to manage arbitrary user-defined data types. On the other hand, a central tenet of an object-oriented data model is the uniform treatment of arbitrary data types as well as the ability to add new data types. Further, an object-oriented data model allows the representation of not only data, and relationships and constraints on the data, as the relational data model does; but also the encapsulation of data and methods that operate on the data.

Another relational shortcoming is the lack of complex modeling which the object-oriented paradigm, through the notions of encapsulation and inheritance (reuse), inherently satisfies by reducing design difficulties and evolving very large and complex databases. The notions of encapsulation and inheritance are a key to further productivity gains in database application development.



Evolution of Object-Oriented Database Management Systems

However, the first generation of object-oriented database management systems were developed to provide transparency for object-oriented programming languages, and were designed initially as single user systems or as file systems with extensions to support concurrent environments. The lack of database management facilities in a first generation OO system is less of a problem in a small, static system. Moreover, first generation object-oriented systems are not able to scale up as the system has to support increasing numbers of users, diverse populations of users, growing data volumes, and high availability. As a persistent object store, a first generation database system fulfills its requirements. But, the requirements of a complete object-oriented database management system are distinguished by its database facilities (generally taken for granted in the relational world) and are fundamental for success for any shareable, reliable database system.

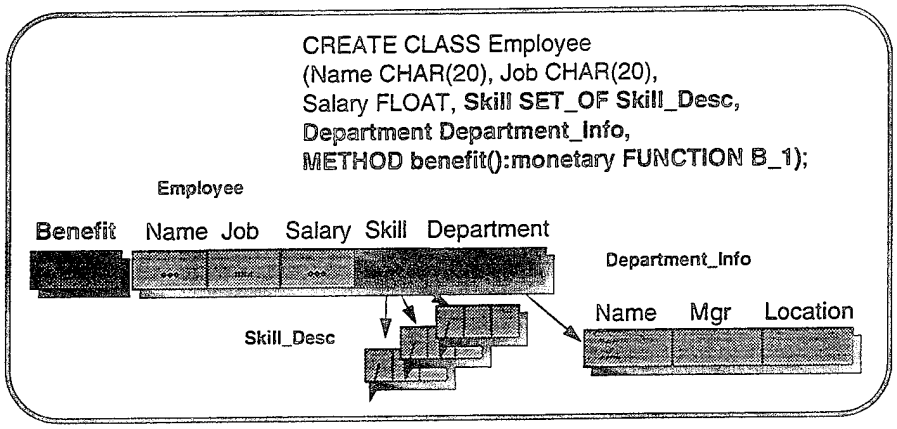
There is little debate today as to what the basic facilities of a database manager include: a DBMS separates the logical from the physical data storage, and automatically optimizes physical storage based on automatically captured statistics (and on "hints" from the database designer in the form of tunable DBMS parameters); a DBMS automatically enforces integrity constraints and cardinality; a DBMS provides a full authorization facility and automatically validates user access to the system; a DBMS provides on-line backup facilities and space recovery utilities, as well as

automatic restart/recovery in the event of a crash; and a DBMS provides dynamic schema evolution to support non-stop mission-critical applications. All of these facilities are built-in, not layered in the interface, i.e., regardless whether access to the database is via Smalltalk, C++ or C or a query language or tool, they are automatic.

The limitations of the first generation of object-oriented data base management systems triggered database researchers to investigate and develop a second generation—object-relational database management systems.

Object-relational DBMSs were designed and built to provide ALL the benefits of the object-oriented paradigm (encapsulation, object identity, inheritance, methods, polymorphism, etc.) plus the best capabilities of pre-relational and relational database technology into a single engine. This architecture provides a formidable array of technical solutions for the most pressing problems faced by information technology organizations:

- pointer chasing to provide the high performance data navigation achieved with pre-relational database management systems.
- complete data and computational models. The object-relational data model supports all features engineered into mature relational database systems to support development of mission-critical applications. These features include full compliance to ANSI/ISO SQL92 database management language with extensions to support the object-oriented paradigm as proposed by ODMG93: updatable views, automatic query optimization, access authorization, dynamic schema evolution, automatic concurrency control, automatic recovery from crashes (including media crashes), triggers, client/server architecture, and more. In actuality, object-relational DBMS goes well beyond most conventional relational database systems in supporting such features as repeating groups, nested tables, etc.



TOTAL ORDB Object Extensions

- a powerful framework for uniform data management and application development support for virtually all types of multimedia data (e.g., text, images, audio, graphics, etc), and even physical or logical devices associated with multimedia applications (e.g., scanners, fax machines, satellite links, video cameras and displays, etc). The multimedia framework includes a built-in class hierarchy of multimedia data types and operations on them. The object-relational model also allows large unstructured data to be stored and managed in native operating system files, just as though they were inside the native object-relational

database. It enables application developers to easily support virtually any type of input, output and storage device as an integral component of the data and application environment. Further, the multimedia framework is fully integrated with the query processing and transaction management components of the object-relational DBMS. Therefore, it supports queries against multimedia data and also maintains integrity for updates against multimedia data.

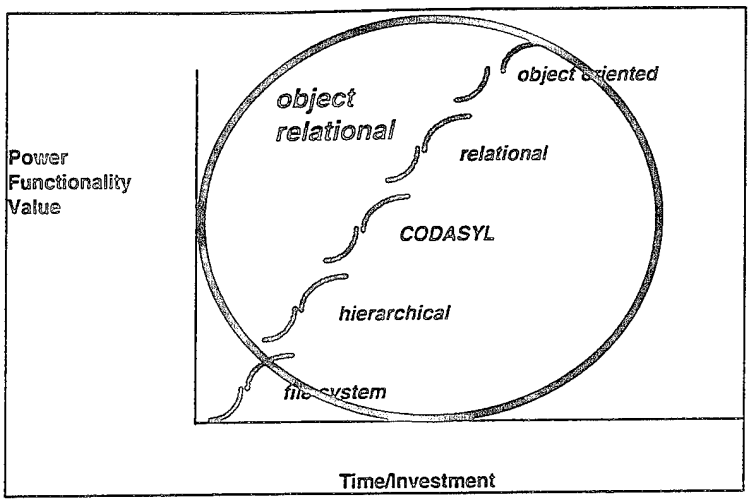
- inherently distributed, multidatabase management environment in which application programs using a single global view and a single database language can access multiple heterogeneous pre-relational, relational and object-oriented databases. Simply put, a distributed multidatabase is a database system that resides unobtrusively on top of existing database and file systems and presents a single database image to its users. It offers a significant improvement in productivity to application developers who develop mission-critical applications that require simultaneous reads and updates to multiple heterogeneous databases. This capability eliminates the need for application developers to use several different database interface languages and to deal laboriously with the schematic differences among multiple heterogeneous databases. Also, application developers no longer need to explicitly manage mutual consistency among multiple databases when simultaneously updating them within a single transaction. All the productivity, performance, multimedia data integration benefits and capabilities of the object-relational model carry over directly to new applicaton development. This means the new object-oriented applications can be built using relational and pre-relational legacy databases.

The technological breakthrough that led to object-relational DBMS translates into numerous benefits for application developers:

First, because the object-relational data model offers a single database language, the application developers do not need to learn and use multiple external database interface languages to develop applications that require access to multiple external databases.

Second, because the single database language of the object-relational model is based on ANSI/ISO SQL92 with object-oriented extensions, productivity of the application developers is almost immediate.

Third, because the data model (and database language of the object-relational DBMS is a natural outgrowth (extension) of the popular relational model and language, the application developers can take advantage of the object-oriented facilities of the object-relational DBMS to be even more productive. The object-relational data model in effect extends external relational databases with object-oriented data modeling and data management facilities.



The Object-Relational model subsumes all previous data models

In summary, the object-relational model was designed to *subsume* the pre-relational, relational and object-oriented models by providing all the inherent capabilities of these models. This approach will provide developers with the capability of developing mission-critical applications in the challenging application domains where today's other database management systems have failed.

U.S. ARMY ARTIFICIAL INTELLIGENT CENTER INTEGRATED DATABASE (AICIDB) SYSTEM

MAJ LEONARD THARPE
CPT(P) DIONYSIS ANNINOS
U.S. ARMY ARTIFICIAL INTELLIGENT CENTER

1. INTRODUCTION

The AICIDB was conceived to provide the Headquarters Department of the Army (HQDA) with an integrated data repository system that consists of a logically unified data model (IDB) and a unified data encyclopedia system (DES). The overall objective of the IDB is to make quality, coherent data widely available and easily accessible to all users. Once the system is fully operational, it will provide the HQDA staff and departmental users desktop access to the IDB and DES.

Currently, to fulfill their information requirements HQDA staff elements must gather data from several different files or databases maintained on different platforms and formats (see Figure 1a.). These "stovepiped" characteristics can be attributed to the fact that they were designed as stand-alone processes with no consideration given for interacting/integrating with other file systems. Therefore, combining the data from these disparate sources to form useful information requires complex and expensive data integration often using outside consultants and software developers. This scenario is played repeatedly across the staff elements resulting in duplication of effort for resource intensive tasks. The IDB will provide a single, centralized, integrated repository of data to eliminate these redundant tasks (Figure 1b.).

The IDB models and stores the synchronized, integrated, and normalized relational data sets that span the key business area within HQDA (unit and organizations, logistics, personnel, facilities, budget, readiness, training, and acquisition). Another feature of the IDB is the Data Encyclopedia System (DES) which provides users with information about the data (metadata) and its linkage to external standard data element (e.g. Army Data Model, C2 Core Data Model). Once the IDB is operational, users will have access to the repository in a client-server environment. The IDB will provide a framework for the development of new systems or the enhancement of existing ones.

The process of integrating Army legacy data into a single data repository presents many interesting challenges, both technical and non-technical. In this paper we present an overview of the Army Artificial Intelligence Center's approach to the development of the IDB for HQDA. The methodology of moving legacy data to the IDB is presented, and the modeling technique is described. The technological approach is presented, discussing the open systems architecture, database engine and components, modeling tools, and data visualization tools. Additionally, research

initiatives that will provide enhancements such as data warehousing, data mining, and intelligent legacy-to-standards data mapping are discussed.

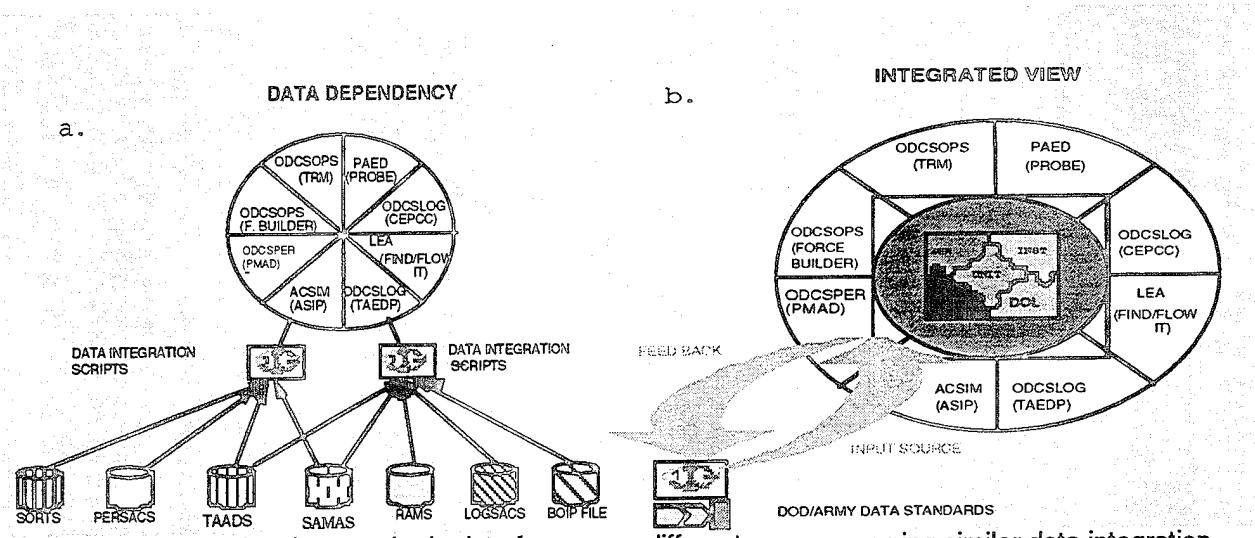


FIGURE 1. a. Functional users obtain data from many different sources running similar data integration scripts. b. Required data can be obtained from one source.

2. METHODOLOGY

Life Cycle Methodology

The AICIDB's objective is to continuously evolve the IDB's integrated data, and to have a DES that is up to date and accurately reflects comprehensive information about the IDB, standard data models, legacy data sets, and mapping. Both the IDB and DES physical data structures are completely managed through their logical data models, and a large percentage of the IDB and DES data population is automated and integrated.

As new legacy data elements are introduced for inclusion into the IDB several steps must occur (see Figure 2.1). First, the legacy data is analyzed to understand its functionality. Existing DOD/Army data models are also analyzed for potential reuse. Next, the existing IDB logical data model is analyzed to identify if new entities or attributes are required. If so, the new elements are integrated into the IDB logical data model in Oracle CASE and ERwin. Once the IDB logical data model modifications are completed, CASE tool utilities are used to auto generate the underlying data definition language (DDL) for the modified physical data structures. After the logical data models and the physical data structures have been modified, the physical data must be populated or repopulated. The IDB data population scripts are then modified to load or reload the appropriate IDB tables with the newly integrated legacy data values.

The DES provides complete and up to date information about what is in the IDB, why it is there, and where it came from. To minimize this maintenance and guarantee its accuracy much of the DES population has been tightly integrated with the CASE tools and IDB data population scripts. All IDB physical data structures are recorded in the DES using IDB developed utilities which pull the current IDB physical structures directly from the CASE data dictionary. In addition, the data population scripts have embedded hooks that record the origin of the physical data values into the DES.

Standard data models (currently DOD Enterprise Data Model, C2 Core, and Army Data Model) are loaded into the DES using ERwin extracts, ASCII extracts, or standard modeling language (SML) exports. ASCII extracts are directly loaded into the DES with IDB developed utilities. ERwin extracts and SML exports are a first imported into ERwin and extracted from the ERwin data dictionary using AICIDB developed utilities.

Legacy data sets are continuously acquired that are related to the force development process or are required by any of the IDB customers. Comprehensive information is captured and recorded in the DES about the legacy data such as its data structures (files, records, and columns), where the data came from, and its authoritative usage (create, associate, pass through, or derived). The exact data structure are recorded so that data loading utilities can be auto generated to minimize legacy data population efforts.

The DES's legacy-to-IDB-to-standards mappings is accomplished using two cross reference entities (legacy to IDB and IDB to standards). To accomplish the legacy to IDB mapping a distinct column list for each legacy system is created and a distinct column/domain list is created for the IDB. This permits mapping of all like legacy data elements to a single IDB column/domain. Similarly, a distinct attribute/domain list is created for each external standard data model also permitting a many-to-one mapping from the standard element to the IDB column/domain. This technique minimizes maintenance and achieves complete legacy to standards mapping.

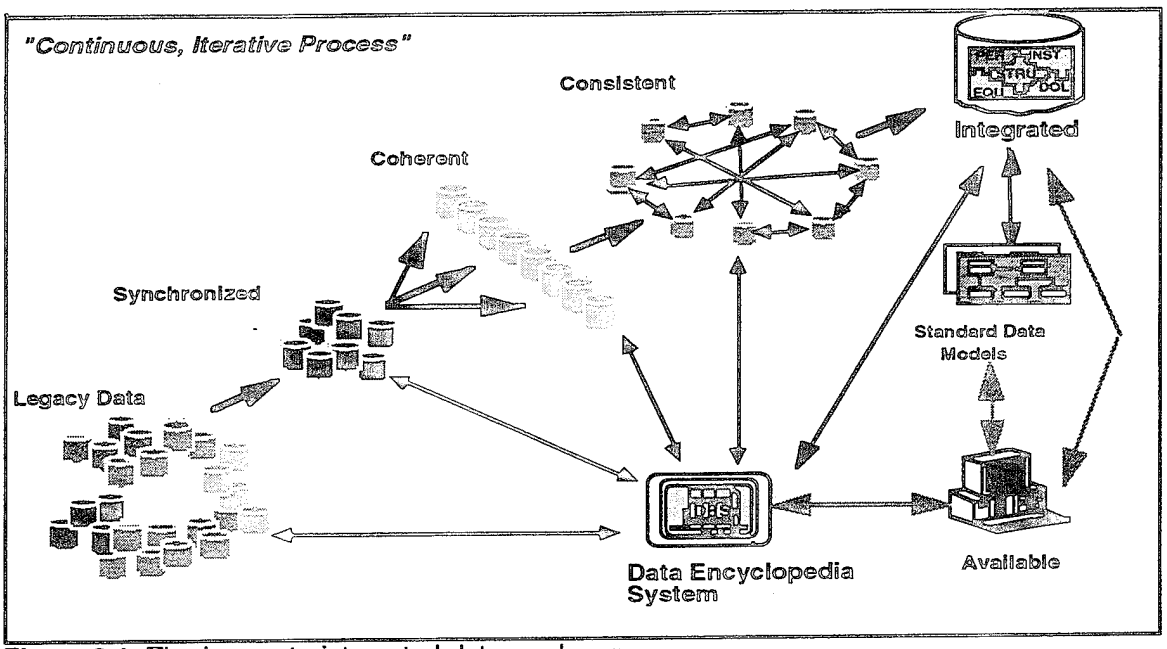


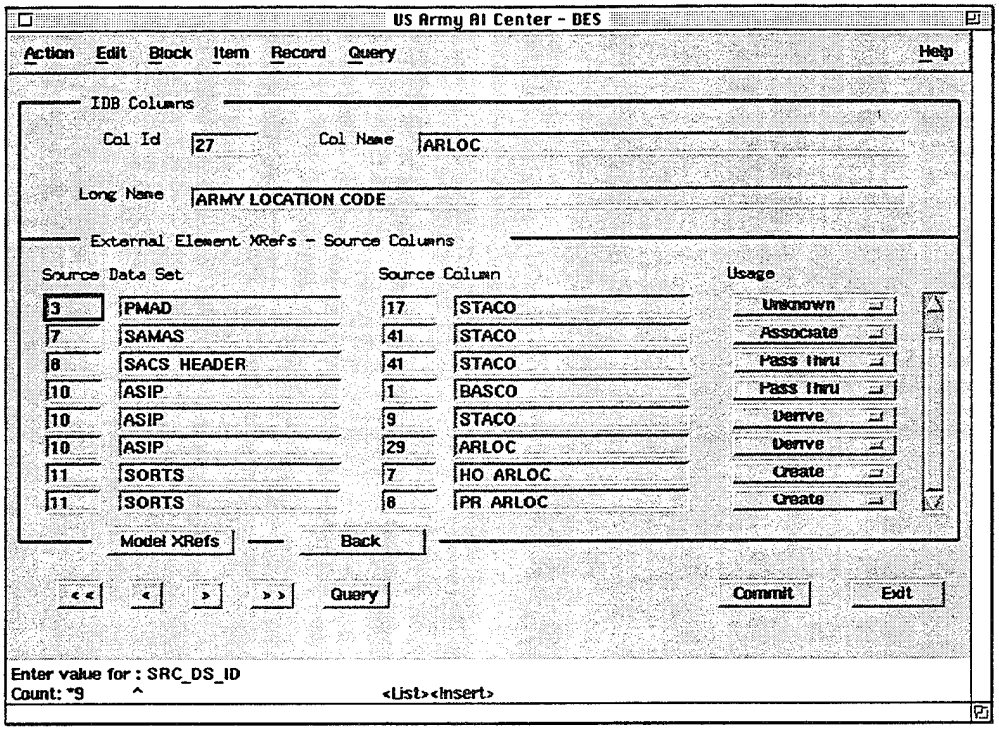
Figure 2.1. The legacy-to-integrated data road map.

Data Encyclopedia System (DES)

The DES provides complete information (metadata) about the IDB data, input legacy data, existing standards, and mappings of the legacy data elements to standard elements. It also provides table and row level security identification, and data acquisition information. In addition to containing knowledge about data, the DES serves as a repository for other Army and DoD data models.

The DES allows users to query for information on each data element in the IDB. Figure 2.2.a shows a DES screen of the Army Location Code (ARLOC) data element. This is a cross-referencing screen that shows the name of every source data set that has this equivalent data element, the column name, and how the data element is used in that particular data set. The DES screen at Figure 2.2.b shows the ARLOC data element and the standard data elements it maps to; Unit Home-Geographical-Location Code in the Army Data Model and Organization-Location identifier in the C2 Core Model.

a.



b.

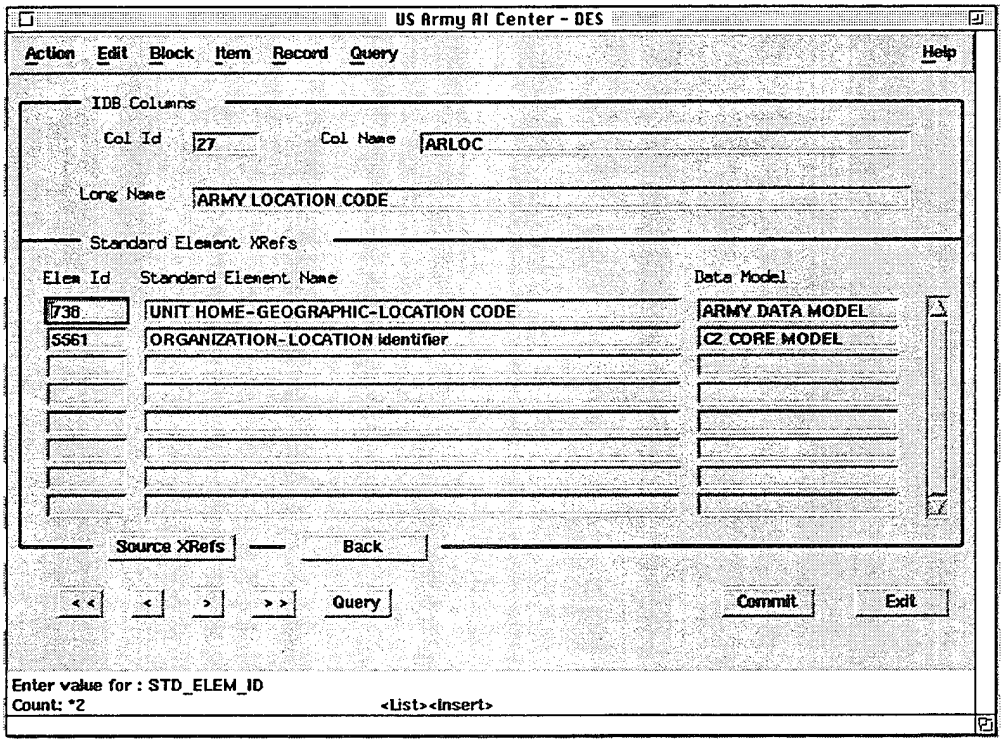


Figure 2.2. DES screens showing (a) data cross-references and (b) standards mapping.

3. ARCHITECTURAL APPROACH

Database Engine and Components

There were four configuration objectives for the IDB. The first objective was to choose a combination of software and hardware that would support database scalability to over 500 gigabytes of on-line storage. The second is to house integrated data in a relational database that supports Structured Query Language (SQL), an industry standard that provides database access to the broadest cross section of software products. The third objective was to maintain performance levels to meet massive processing demands when batch updates, on line transaction processing (OLTP), and backups compete for resources as the database grows. The forth goal was to provide 7-day, 24-hour availability.

ORACLE 7 was chosen as the IDB RDBMS for many reasons. ORACLE has had proven success with very large databases (VLDBs). The RDBMS is designed to maximize performance on symmetric multiprocessor processing (SMP) and massively parallel processing (MPP) UNIX platforms. ORACLE Parallel Query technology spreads operations like queries, indexing, and loading across multiple server processes which in turn are spread across multiple processors. As the IDB scales into the hundreds of gigabytes, performance features will be critical. ORACLE also supports ANSI standard SQL, and most major software vendors offer connectivity to ORACLE. Version 7 of the RDBMS incorporates advanced mechanisms for enforcing data integrity and consistency, including integrity constraints and database triggers. Also included is support for truly distributed databases, with data residing on multiple nodes in separate physical locations, while maintaining data integrity and ORACLE 7 includes the powerful procedural language PL/SQL, which adds traditional third generation language (3GL) constructs to SQL. Finally, ORACLE offers advanced computer- aided software engineering (CASE) tools that are highly integrated with Oracle application building products such as ORACLE*Forms and ORACLE*Reports.

Oracle's latest announcement, ORACLE VLM, will support very large memory configurations (2 gigabytes or more) on 64-bit machines such as the DEC Alpha. The combination of current and projected features made ORACLE 7 the product of choice for the IDB.

The IDB uses ORACLE 7.1.6, with Procedural and Parallel Query options, running on a Sun SPARCcenter 2000 under Solaris 2.3. The 2000 has eight processors and 1 gigabyte of RAM. Attached to the server are five non-RAID drive towers with four drives each. The towers house eight, 2.3-gigabyte drives, and twelve, 1.5-gigabyte drives. Each tower has one controller in the host system. Client workstations such as Sun SPARCstations and Windows PCs are networked to the server via a fiber data distributed interface (FDDI) channel.

Twenty-four gigabytes of disk space have been allocated for the IDB, of which a little over 10 have been used. The underlying data files, log files, and control files have been balanced across all twenty disks. The directory structure, naming conventions, and I/O distribution follow the Oracle Optimal Flexible Architecture (OFA) standard. OFA is designed to promote administration ease by allowing fast reconfiguration of growing or changing databases.

The ORACLE 7 system global area (SGA), a collection of memory buffers that hold data used to resolve SQL operations, uses 632 megabytes of contiguous shared memory. Within the SGA, 440 MB are reserved for the database buffer cache, which is the primary storage area for data retrieved by SQL statements. The IDB database is currently configured to handle heavy batch data manipulation activity rather than OLTP. Archivelog mode is not currently used because, in addition to nightly full backups, we can rebuild data from load files with the same amount of effort as recovering archived log files. This allows us to avoid additional I/O on one of five controllers. Our Redo Log files are sized at 400 megabytes each to approximate the database buffer cache size; this minimizes checkpoints due to log switches. Also, the redo logs are not multiplexed. The large SGA and balanced I/O distribution maintain high performance during heavy activity periods. Following the OFA standard allows us to flexibly reconfigure in order to maintain performance as the database grows.

Planned enhancements to the IDB installation include a Data General Clariion RAID consisting of 88 gigabytes of storage, as well as multiple digital linear tape (DLT) tape drives, or a DLT RAID, to backup the growing database. Eventually, the IDB may be migrated to a 64-bit platform such as the DEC Alpha using ORACLE VLM. The IDB may also employ the ORACLE 7 Distributed Option in the future to provide the most effective access by agencies that make up the user population.

Planning a database installation that will scale to 500 gigabytes or more forces preparation for contingency as much as current reality. Many things may change over time as users creatively employ IDB data. It remains apparent that the foundation of preparation and implementation of this VLDB is database software that is designed to meet that task. The ORACLE RDBMS has been optimized to take advantage of the latest technology available; and new Oracle developments such as ORACLE VLM indicate that the software will be optimized for future technology as well. The RDBMS has become the cornerstone of IDB development for that reason.

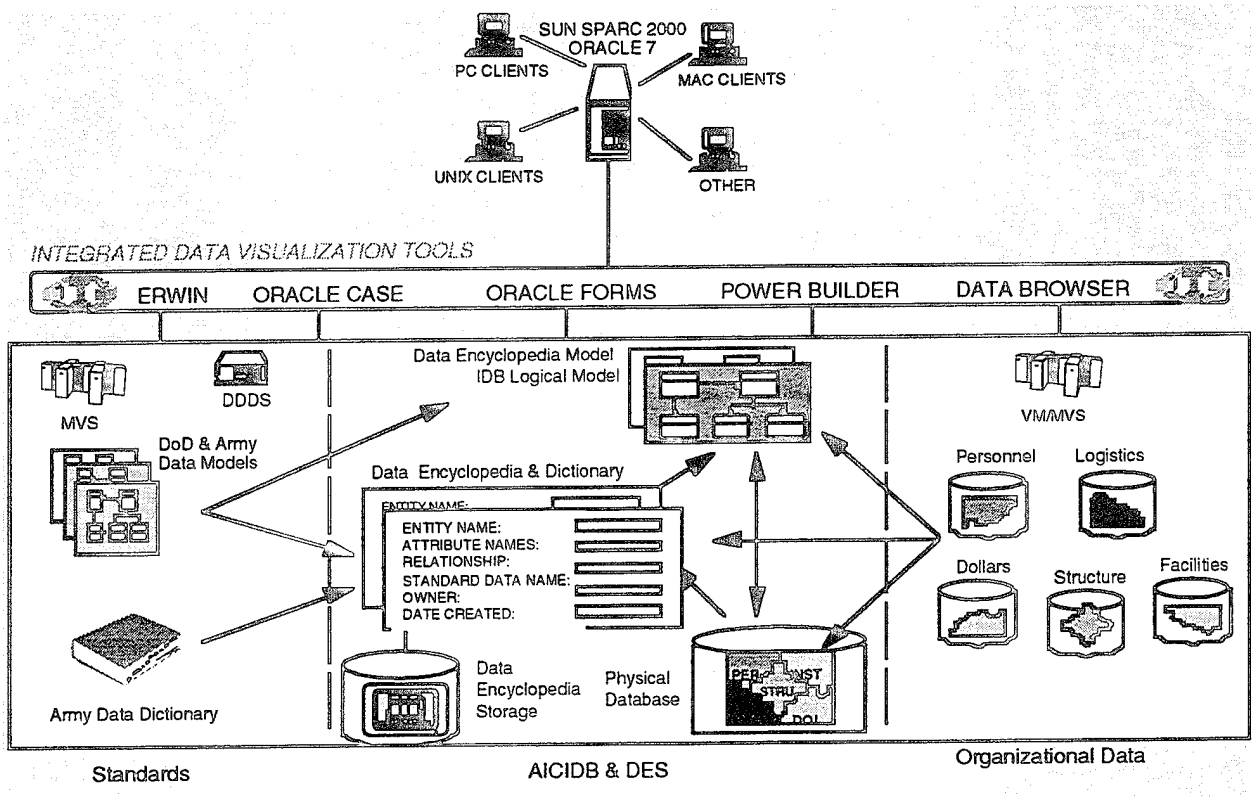


Figure 3.1. Architectural overview of IDB and DES.

Modeling and Data Accessing Tools

Oracle CASE and Logic Works ERwin are being used to build the entity-relationship (ER) diagrams. These diagrams provide the required knowledge base that explicitly represents real world objects and their relationship among other objects. Both tools allow for complete management of the database from the conceptual model (ER diagram) to the low level model where the physical data is stored. The ERwin product gives us the ability to produce models that are in compliance with the DoD standard for data/information modeling, Integration Definition for Information Modeling (IDEF1X). It is also used to converting the models constructed using Oracle Case to IDEF1X compliant models, using the ERX for Oracle Case version.

The data accessing tools currently being used to access data in the IDB and DES are Oracle Forms 4.0, and 4.5, and Powerbuilder 4.0. Each of these provides a user-friendly graphical user interface (GUI) with point-and-click functionality. Oracle Forms 4.0 is used in the UNIX environment, Oracle Forms 4.5 is used for access via PCs, and the Powerbuilder tools provides PC access in a client-server environment. Figures 3.2.a and 3.2.b show GUI screens designed using Oracle Forms 4.0. Figure 3.2.a is used for querying and viewing information for a particular unit or organization. The design and layout of the screen allow the user to query on several different fields. The buttons on the screen allows the user to either access information for a different unit or go to a totally different screen simply by using a mouse. Figure 3.2.b shows similar features for accessing information on equipment.

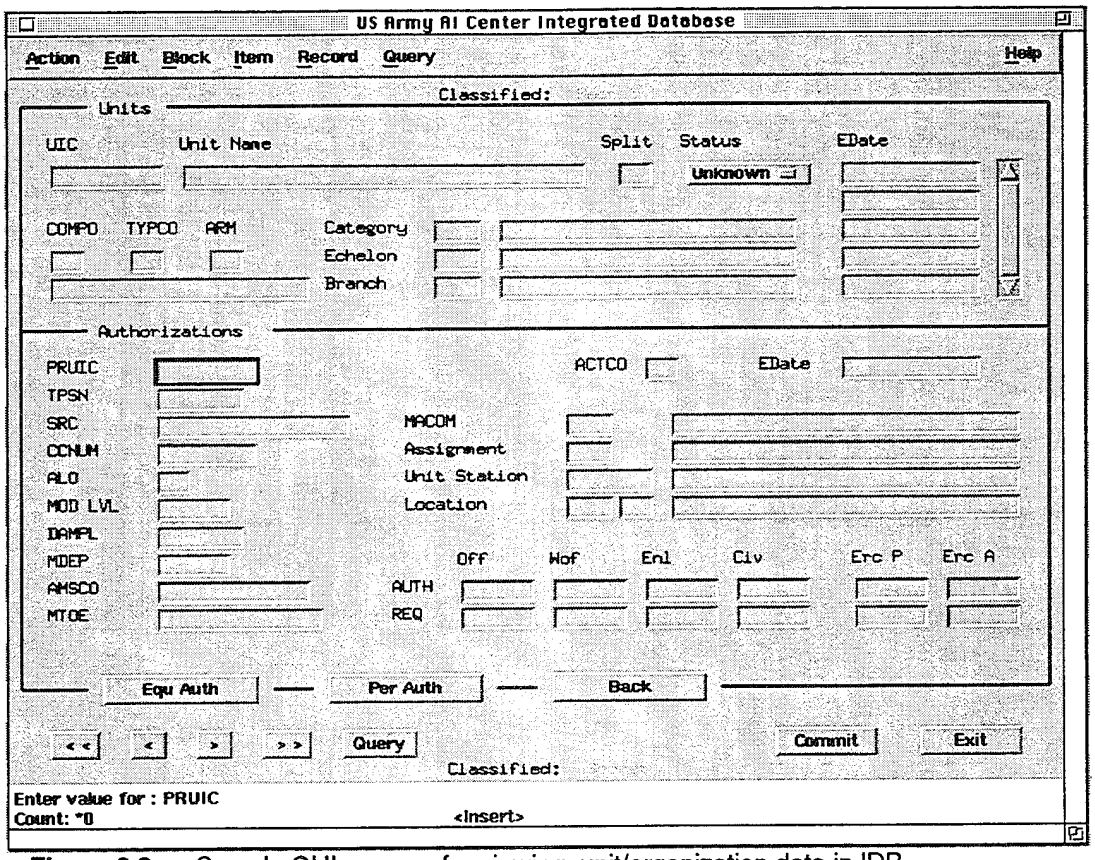


Figure 3.2.a. Sample GUI screens for viewing unit/organization data in IDB.

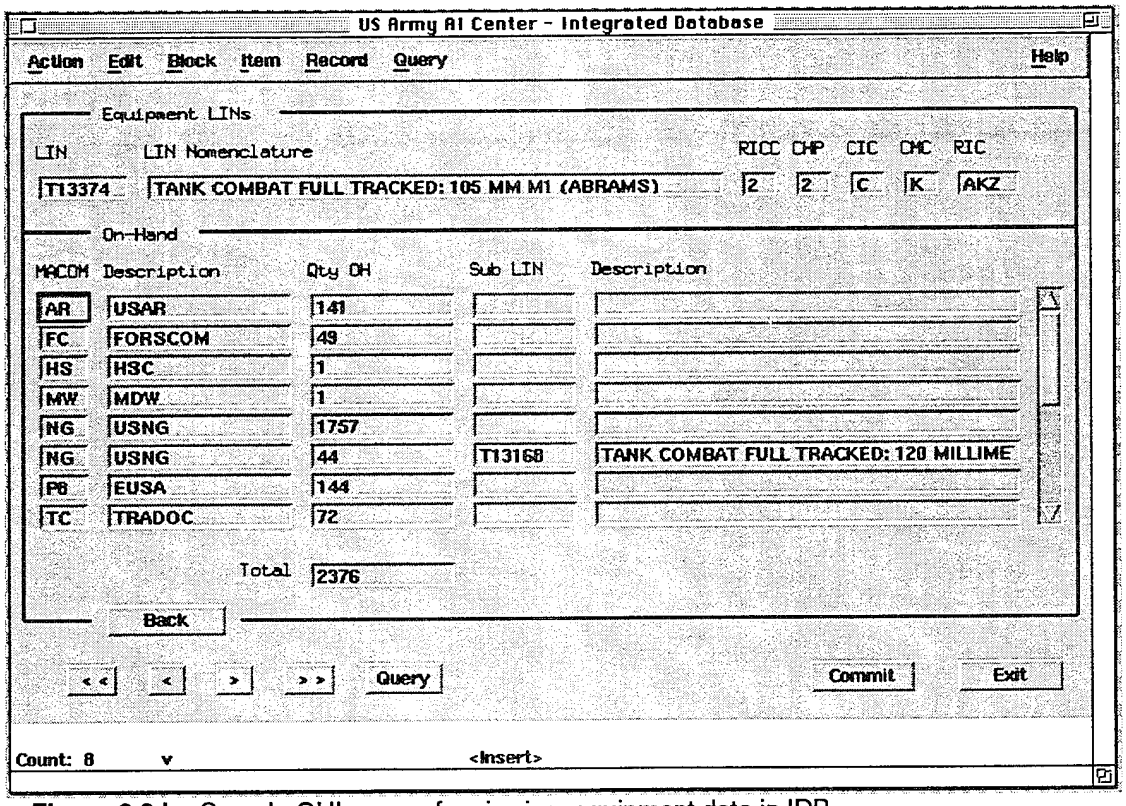


Figure 3.2.b. Sample GUI screen for viewing equipment data in IDB.

4. ENHANCING THE IDB

Once the prototype of the IDB reaches a predetermined and validated degree of functionality, it will be handed over to the HQDA Information Management Center's (IMCEN) Data Management Team for implementation and management. However, the AI Center will continue to pursue state-of-the-art technology to further advance and improve the capabilities of the IDB. These enhancements will initially be targeted towards data analysis tools and intelligent data accessing in a client-server environment. The first planned enhancement is to evolve the IDB into a comprehensive data warehouse. The purpose of data warehouse is to better meet the unique data requirements of users of decision support systems, for high level views of the data for management, and for performing complex analysis.

Another enhancement to be added is data mining tools. These tools will allow data managers and analysts to identify specific patterns of information in the data. This is particularly useful for examining historical data to make predictions or decisions concerning future events. The AIC is also investigating a tool that provides an intelligent interface that allows users to perform ad hoc queries without knowledge of the underlying structure of the database or relationships between the tables. The users' queries need only contain attribute names and conditions. Other areas of research interests include an intelligent data mapping capability that will provide a more efficient and effective means for mapping legacy data elements to standard data elements, and the possibility of providing services to World Wide Web (WWW) clients.

5. BENEFITS OF THE IDB

The IDB offers many benefits to users of all levels of the HQDA. It provides a single, integrated source for Army data. Users will no longer have to collect the data from the many different legacy systems and platforms. This also means that departments may realize considerable savings in time, money, and other resources because they do not have to develop the complex data integration applications when using the legacy data. The modeling and integration methodology used in bringing the data into the IDB provides users with a synchronized, integrated view of the data that reflects the key functional or business areas of HQDA. The system also assists in improving the quality of the data brought into the IDB by providing feedback to the original sources concerning data discrepancies, rule violations, and other data errors.

One of the benefits of the DES is that it provides an extensive amount of pertinent information about the data contained in the IDB. Thus, the users do not have to guess about the meaning of a particular data element, its origin, or what standard

data element it maps to. This is particularly useful in the development of new systems and migration of existing systems to the IDB. Also, because of the open systems design approach, users will be able to access the IDB and DES using their current desktop computers (PC, MAC, UNIX, etc.) and a variety of data access and visualization tools. The source data owners also benefits from the IDB and DES because of the feedback they receive from the system concerning data discrepancies. This will assist them in improving the quality of the data they provide. Figure 5 lists other benefits provided by the IDB.

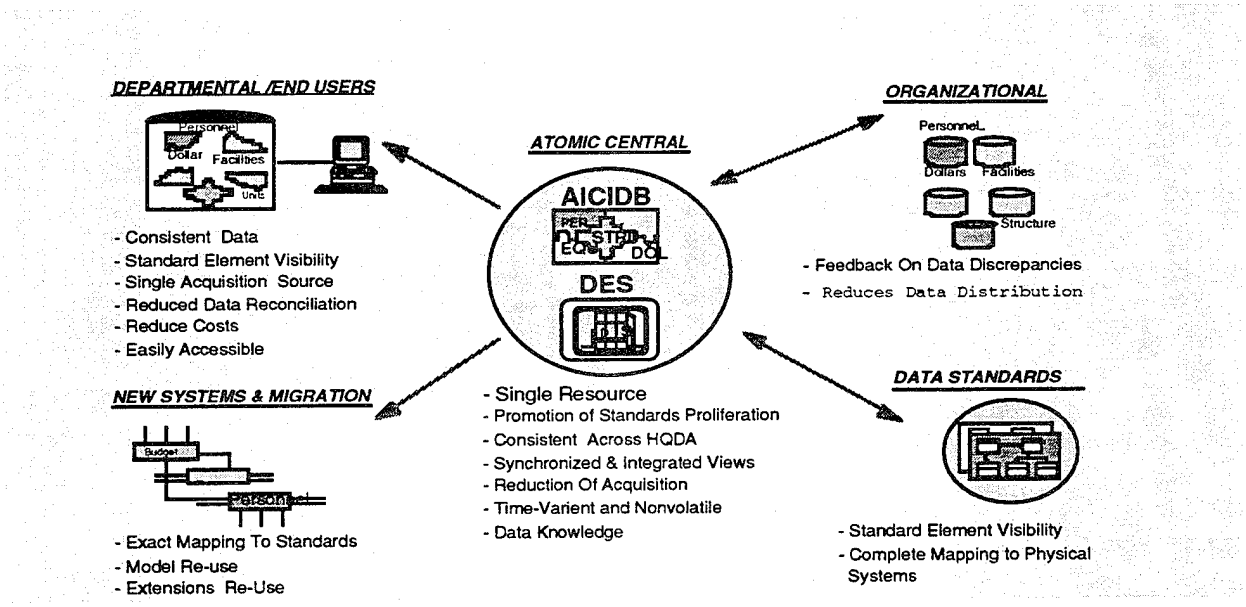


FIGURE 5. The potential benefits of the IDB.

6. SUMMARY AND CONCLUSION

Planning and developing the IDB has not been without its challenges and tough issues. The issues of defining the scope of the project and determining the user requirements present a challenging situation. Generally, the scope is determined by the requirements. However, sometimes it is difficult for users to determine what they need until they see what the system is capable of providing them. The process becomes iterative and continuous as the users requirements are refined or changes. As additional functionality is added to the IDB prototype, it will ideally evolve into the fully functional system. It is realized that a product, perhaps an early version of the target system, must be delivered to the users within a specified time frame, otherwise they will lose interest and see this as another futile attempt at solving a data management problem.

There is also the issue of data ownership. Data owners need to be assured that they are not giving up the ownership or control of the data but making it available to the IDB for all valid users. The users will not be allowed to modify the data or make

changes to any of the source tables in the IDB. The owners will still be responsible for the maintenance and control of their data.

Army decision-makers and analysts rely heavily on the available data in making important decisions on a daily basis. Often there is not much lead time given to make such decisions, so the required data must be readily available and easily accessible. Also, the quality of the data available to them will most likely affect the quality of their decisions. Providing an IDB environment as presented in the sections on methodology and technological approach, and, eventually, the envisioned enhancements described in that section, should go a long way in meeting the information requirements of the HQDA staff, providing the benefits described in Section 5.

Authors and Contributors

MAJ Leonard Tharpe is the head of the IDB Team in the US Army AI Center's Knowledge Engineering Group. He is an Acquisitions/Systems Automation officer. He holds a Masters Degree in Computer Science from the U.S. Naval Postgraduate School. He has been with the AI Center since March 1995. His address is U.S. Army Artificial Intelligence Center, ATTN: SAIS-AI, 107 Army Pentagon Room 1D649, Washington, DC 20310-0200. Phone, (703) 697-6578 DSN 227-6578 (email: tharpe@pentagon-ai.army.mil).

CPT(P) Dionysios Anninos is currently attending the Army Command and General Staff College in Ft. Leavenworth, KS. He was previously the head of the IDB Team. He is an Engineer/ORSA officer and holds Master Degrees in Systems Engineering and in Transportation Engineering from the University of Pennsylvania.

Mr. Rick Martin is the Senior Database Design Engineer and head systems architect of Leading Technology Services Corporation, a subcontractor for NCI Information Systems. He has worked on the IDB project since the initial stage.

Mr. James Froio is an employee of NCI Information Systems in McLean, VA. He is the Database Administrator (DBA) for three Oracle installations at the Army AI Center. He has over 7 years experience with Oracle DBMS.

Final Report of The DBSSG Predictable Real-time Information Systems Task Group

Donna Fisher
NCCOSC
SanDiego, Ca 92152-5001
Dfisher@cod.nosc.mil

Paul J. Fortier
University of Mass. Dartmouth
North Dartmouth, Mass. 02747-2300
Pfortier@umassd.edu

David K. Hughes
DBx inc.
CherryHill, NJ 08002-0446
DBX@world.std.com

Mayford Roark
Martin Marietta
Syracuse, NY 13221-4840
Roark@gw.syr.ge.com

1. Introduction and Goals of the PRIS-TG

The Predictable Real-time Information Systems Task Group (PRIS-TG) is a task group of the X3 (Computers and Information Processing Committee) Database Systems Study Group (DBSSG). PRISTG's relationship to the X3 standards organization is described below. The purpose of the PRIS-TG is to determine if the technology for real-time systems in general and for real-time information management is sufficiently mature for standardization.

The PRIS-TG was tasked to develop a Predictable Real-time Information Systems Reference Model, evaluate existing Predictable Real-time Information Systems technology, and determine the need for standardization in this area.

The objective of the PRIS-TG study was to establish a framework for future predictable real-time information management standards activities, both extensions to ongoing SQL (Structured Query Language), IRDS (Information Resources Dictionary System), RDA (Remote Data Access), ODP (Open Distributed Processing), OIM (Object Information Management), POSIX (Portable Operating Systems Interface for Computer Environments), Ada, and computer security development, as well as related future standards.

The task group reviewed and evaluated existing and developmental products claiming to be real-time information management systems, as well as, real-time products with information management capabilities. The task group also reviewed and evaluated published literature and research activities concerning real-time information management technology world wide. In this report the PRIS-TG will discuss the recommendations for standardization in the area of real-time information management technology.

1.1 Overview of Document

Beyond this introduction section, this document is broken up into three additional sections. The section 2 indicates the issues with conventional database technology when applied to real-time systems. Section 3 outlines PRIS-TG findings on the state of real time database technology and which technologies need to be addressed to realize real time database management systems and standards in particular. Section 4 indicates PRIS TG recommendations for X3 work towards real-time database management systems standards. Further details on real-time database systems is contained in the PRIS-TG real-time database management reference model[For93].

To highlight the needs and show where work must progress, an annotated bibliography is included on papers that have been generated by PRIS-TG members and other organizations on real-time database topics of interest. These papers include works on:

flexible transaction structure and specifications, real-time structured query language concepts, semantic database management systems models, other ongoing standards activities in real-time database systems, the operating system and real-time database management interface issues and real-time scheduling and evaluations. These represent just a sampling of ongoing efforts in real-time database management.

1.2Terminology Used

The PRIS-TG uses the following terms and meanings within this document.

- Data Item - the smallest separable unit recognized by the database representing a real-world entity.
- Database - a collection of data items which have constraints, relationships and a schema.
- Schema - a description of how data, relationships and constraints are organized for user application program access.
- Constraint - a predicate that defines all correct states of the database.
- Transaction - a partially ordered sequence of database operations that represent a logical unit of work and which access a shared database.
- Transaction Properties - ACID properties are defined as:
  - -Atomicity - either all the actions of a transaction occur successfully or the transaction is nullified by rolling back all updates.
  - -Consistency - a transaction moves an object ... from one valid state to another valid state,and if the transaction is aborted, the object is returned to its previous valid state.
  - -Isolation - the actions carried out by a transaction against a shared database cannot become visible to other transactions until the transaction commits.
  - -Durability - once a transaction completes successfully, its effects cannot be altered without running a compensating transaction. The changes made by a successful transaction survive subsequent failures of the system.

- Relationship - a correspondence among two or more data items.
- Features of data items in a database
  - -Shared - data items in a database are shared among several users and applications programs.
  - -Persistent - data items in a database exists beyond the scope of the process that created it, the data exists permanently.
  - -Security - data items in a database are protected from unauthorized disclosure, alteration or destruction.
  - -Validity - also referred to as data integrity or correctness. Data items in a database should be correct with respect to the real-world entity that it represents.
  - -Consistency - whenever more than one data item in a database represents related real world values, the values should be consistent with respect to a defined relationship.
  - -Non-redundancy - no two data items in a database represent the same real-world entity.
  - -Independence - data items in the database should exhibit physical data independence and logical data independence.
- Concurrency control - the activity of coordinating actions of concurrently executing transactions so that the correctness of the database is maintained and transaction properties are not violated.
- Recovery - the activity of the database management system that provides restoration of the database when transactions fail. The affects of aborted or failed transactions must be isolated to only failed transactions no others should be affected.

- Real-Time - that area of computer systems design and implementation in which the correctness of a result is dependent on the validity and timeliness of data and of the operations on that data.
- Real-Time Database Management System - a database management system which is capable of managing time-constrained queries. Such a DBMS may form the foundation in support of time-constrained transactions.
- Real-Time Transaction - a complex query made up of more than one action that must be performed within a given period of time. The transaction must be completed or aborted as a "single unit".

2. Real-Time DBMS Problem Statement

Real-time information processing requirements are found in a wide spectrum of applications, from small embedded systems to large transaction oriented systems. The common factors amongst these systems is the need for determinism and high speed in information accesses and manipulations. The need for *real-time database management systems* is becoming more evident as large data intensive projects such as the National Information Infrastructure (NII, or information highway), are being developed. NII applications include electronic commerce, digital libraries, advanced manufacturing, environmental monitoring and health care. Beyond NII applications other real-time database application areas include telephone, energy management, automated factories, airplane information management, defense oriented systems, prison management, cable television, medical and financial management. These applications require application derived data structures, high data availability and time constrained access to data that may also have time constraints on data consistency and correctness. Requirements for real-time database management systems encompass features for database and transaction structure, transaction execution policies and architectural interaction requirements with the system's infrastructure \cite{Gordon:requirements}.

Applications such as those above require a different model of database structure and database processing than that found in conventional database management systems. Adherence to the conventional database model correctness criteria based on transactions executing to preserve *ACIDity* (Transaction **ACID** properties guarantee the **A**tomic, **C**onsistent, **I**ssolated and **D**urable execution of transactions on the database which the database management maintains in a consistent and correct), properties must be altered if real-time service is to be provided. Real-time does not simply imply *FAST*, but timeliness and predictability in all aspects of transaction and database operations. The need for new solutions for real-time data storage and processing has spurred real-time database systems research. In recent years, significant research has been conducted to develop real-time database models [28, 29], real-time transaction scheduling [2, 11], real-time concurrency control [6, 39], real-time transaction structuring [45, 10] and real-time database recovery [13].

Presently there is a trend in industry to develop products that conform to open systems standards. Database management systems are no exception, though standards in this technology area are relatively new and still evolving. Presently there are real-time database products available (DBx inc, Martin Marrietta, Westinghouse). These products do not yet address all the needs of real-time database applications, though they are a step in the right direction. These products however, do not conform to real-time database standards as none exist. These products have concepts for time driven transactions, time constraints on data and some architectural dependency considerations. Additionally they interact with the hardware systems through real-time open system operating systems standards such as POSIX 1003.4 [3].

3. Findings of the PRIS-TG

The Predictable Real-time Systems task group (PRIS-TG) performed a study to determine the maturity and viability of real-time database management systems to include such technology in existing and emerging ANSI standards. The findings highlight the lack of support for real-time in conventional database systems, the need for such support, the maturity of real-time database technology, the availability of products and supporting standards. Presented are synopsis for recommended real-time concepts for database management systems standardization.

3.1 Real-time Database Systems Technology

Nearly all existing database standards and existing products do not have support for real time service to applications. Conventional SQL products are based on the model of a monolithic database (persistent storage) with serialization of transactions acting on the database as the correctness criteria. Conventional database recovery is based on the check pointing of the database with the application of log actions to redo or undo the effects of transactions on the database at the time of a failure.

While conventional database management implementations support a wide range of application areas, it has been shown by numerous researchers, product developers and vendors that this model limits the ability of databases to be applied to a wider array of computer based information processing and management applications such as those for real-time computing systems. Numerous examples of how real-time database management could improve performance within the computer aided design, computer aided manufacturing, medical monitoring and diagnostics, department of defense mission critical systems, on-line transaction processing systems and numerous other applications areas have been defined in the literature [19].

Work has progressed beyond the research phase into testbed systems and currently into available off the shelf products that exhibit features necessary for the support of real-time in the database environment. These products and research address the need to extend conventional database concepts to include time as a component for database consistency, correctness and manipulation and to address the requirement for predictability of access. In the database specification area the move is to limit the flexibility of on-line alteration of database structures to deliver predictable access to the database while giving more control to the database designer to limit how the database can be used, where it will be stored, how it will be structured in storage, how the database is partitioned and how constraints affect correctness and consistency.

Research, prototyping efforts and product developments address the need of the real-time programmer to exhibit more control over the specification, structure, access, manipulation and recovery of the database and transactions. The focus of these efforts is on the loosening of the conventional *ACID* properties defined for a database's transactions and the development of more flexible transactions to increase concurrency, increase data availability, limit data blocking, not cause cascading aborts and exhibit controlled recovery all under transaction control [5, 8, 10].

Standards in other areas of an information infrastructure which support real-time are also evolving. The IEEE has recently released a real-time extension (IEEE 1003.1b) to the POSIX operating system interface standard (IEEE 1003.1). This standard provides support for real-time scheduling of tasks, the concurrent execution of tasks and for extended control over numerous elements of the computer system. Evolving SQL standards are looking into the support of concepts which will aid in the introduction of

real-time into the standard [10, 12]. For example there are additions in the SQL-3 standard for objects, triggers\footnote{The concept for triggers presently in SQL are sketchy at best} , time reference, more refined constraint definitions and enhanced database and transaction structuring [15, 24].

When taken together these indicators demonstrate the need for real-time databases and the viability of the technology. Real-time database management systems will become commonplace in the computing arena in next five to ten years. To ensure that the development of such systems result in interoperable, open products requires standardization of application interfaces. It is a recommendation of the PRIS task group that the technology is sufficiently mature (concepts well understood) that the standardization efforts should begin now. The next section summarizes some of the findings of this task group for work areas for extensions to existing and evolving ANSI and ISO data management standards.

3.2 Work Areas for Real-time DBMS Standardization

If real-time is to find its way into the present and evolving SQL standards for databases, a working group must be formed to refine concepts defined by the DBSSG PRIS-TG as required for real-time database management systems support. In particular the PRIS-TG has defined five (5) major functional areas within a database management system that need enhancement for real-time to be incorporated into the SQL3 standard. These areas are:

1. Database Structure
2. Transaction Structure and Operational Properties
3. Transaction and Database Recovery
4. Architectural Dependencies
5. Remote Data Access

Each of these will be briefly covered in this executive summary section leaving details for the referenced documents.

3.2.1 Database Structuring. As in all database applications, the writer of a real-time application should be able to access data from the database in a natural form for the application. While the two dimensional table object type of the relational model is very well suited for most conventional applications, it is often insufficient for real-time applications [1, 28]. Relational tables are a poor fit for such real-time applications as contour maps, satellite images, and sensor data. For real-time applications, a database management system must also support the definition of abstract data types and binary large objects (blobs), user specified database consistency constraints, (data type constraints, temporal constraints, spatial constraints, storage constraints, access constraints, data and transaction criticality), as well as database decomposition and distribution.

3.2.3 Transaction Structure and Operational Properties. Transaction structuring must be altered from the conventional straight line sequence of database operations to provide real-time services tailored to the needs of real-time applications [5, 11]. For example if a real-time process is monitoring some number of sensors, a transaction supporting that task must be update data in support of the task regardless of all other actions in the system. Application-dependent transactions may require the ability to specify and control a variety of transaction processing schemes such as; serial transaction partitions [31], nested transaction partitions [25], interleaved transaction partitions [7] and parallel transaction partitions [27]. A general model of a transaction may consist of boundary markers, a specification, a body, recovery handlers and pre and post conditions on

execution. The specification provides data structure definition, timing requirements specification, resource limits specifification (e.g. max CPU, Disk, I/O execution time limits), data dependencies, criticality of transaction, atomicity of transaction, premptability of transaction and execution dependencies definition. Transaction pre conditions and post-conditions are predicates which define the conditions upon which this transaction should or must begin execution and conditions upon which this transaction's execution is considered correct. The recovery body is used to hold user or system defined recovery procedures for user or system specified failures [5, 10, 13].

A new paradigm for transaction execution and concurrency control is needed to support the needs of real-time applications [4, 8, 16, 34, 37]. Transactions must execute to maximize the timeliness and predictability of applications, yet must also maintain database consistency and correctness. The difference from conventional databases is an altered definition of transaction *ACIDity* properties. Real-time transactions must be able to specify their own consistency, correctness criteria, database and transaction partitions, and commit criteria. Transaction initiation and transaction operations concurrency management must be driven by applications temporal, data dependencies and transaction relationships [38]. Transaction writers must be able to control how a transaction is selected for invocation [26], if a transaction can be preempted, if a transaction must be atomic, if a transaction must be recoverable, if data manipulation will trigger other database actions, how a transaction is partitioned and how interleaved conflicting operations will be ordered [10,18].

3.2.3 Transaction and Database Recovery. Conventional means for recovery of committed and active transactions use check pointing of data items, with redo for committed and undo for active transactions. This model of recovery is not adequate for real-time database management systems where availability and timeliness of data may be more important than strict serializability [5, 36]. Correctness criteria for transaction execution and recovery must be altered to support the unique needs of real-time databases [13, 32]. It may be more desirable in the real-time database environment to do nothing (e.g. delay and wait for the next periodic update), do an application dependent recovery, or recover to a future correct state [5, 13] using forward recovery techniques.

Transaction recovery policies and mechanisms for real-time need to be added to the specification of database's data definition language (data temporal consistency constraints and enforcement rules) and to the specification of a database's data manipulation language for transaction specification (transaction temporal consistency constraint and enforcement rules). The ability of transaction writers to specify exception conditions for software transaction aborts, conflict resolution, and hardware faults must be added to database languages [10].

Additional recovery mechanisms for recovery from failures for memory-resident database management systems are required, for bounded recovery in the event of catastrophic failure, the manage the effect of distributed architectures on recovery in real-time, and to the semantics of time-bound recovery (partial recovery, discard and reinitialize, etc.) must be provided within a standard for real-time databases.

3.2.4 Architectural Dependencies. Real-time systems must interact with an environment that they do not control. A real-time system must respond to detected conditions within time frames defined by the physical system so as to affect applications operations in a predictable manner. To provide this the real-time computer systems operations must be totally defined and bounded. To predict the actions of the real-time computer system requires the bounding of all aspects of execution, and to control performance so that actions {\em always} behave in the same way and take the same bounded time.

To deliver such capability the database system must request the operating system to perform in specific fashions. For example, to bound execution time may require limiting the size of a database table and to force the table to be stored in a specific place in memory and stored in a particular fashion. In addition, the database may require certain external sensors to trigger database actions on particular boundaries of time or events outside of the domain of the database system. To limit how the database executes may require the specification of CPU time, I/O time, stack usage and numerous other formerly operating system controlled resources to be managed for the databases purposes.

3.2.5 Remote Data Access. The I/O in a real-time system is not limited to memory and disks. It also includes the network and external elements. These external elements must be properly integrated with any database management scheduler scheme. Gigabit networking technology is making distributed real-time systems possible. Real-time scheduling of data packet/message transmissions will allow for dynamic selection and correlation of distributed data across the system.

Monolithic database management will be more of the exception than the norm in the future. Client/Server and distributed systems are necessary to meet the requirements of next generation products (as seen in the down sizing of corporate databases). Across these distributed components transactions management scheme should be able to maintain temporal constraint requirements as well as access timing requirements accross remote access conduits and protocols.

Realtime systems are subject to an overall system design. They are dedicated to a single application, which may consist of one or more concurrent processes. Resources are still managed by the operating system, but at the direction of the applicaiton, and in accordance with the overall system design. This is especially true of large real-time systems, such as the US Navy's BSY-2 system and the Air Force's AWACS programs.

Providing a framework of integration will make for a more unified approach to developing a real-time system. It also helps foster use of standards and less hand coding by providing mechanisms to control other performance components. The remote data access protocol standards [21, 22] are examples of standards for remote access of data which provide such a framework.

4. Recommendations of the PRIS-TG

The X3/OMC DBSSG PRIS-TG has found that real-time data management technology has a solid foundation for standardization. At the present time there are existing products [20, 30] which exhibit fundamental features for real-time data management. In addition numerous governmental, industrial and academic research programs are developing further prototypes and refinements of the numerous areas mentioned in this report. For example, the University of Massachusetts Amherst has developed real-time concepts, and prototypes for scheduling protocols, real-time transaction processing and real-time recovery; the University of Rhode Island, University of Virginia and Carnegie-Mellon University have been researching real-time database management systems and constructing testbed systems, the US Air Force's FIRM (Functionally Integrated Resource Management) program and numerous others have established a solid foundation for real time data management standards development.

The technology is at a point where many basic concepts have reached stability as indicated by the increase in product developments. The further refinement of these basic features will enhance the stability of any developed standard. Some areas within this

technology will require addition work to refine and stabilize the concepts to a point where a specific and lasting standard could be developed. In the time frame it will take to develop a standard, current and potential advances in the real-time database management area should have matured and stabilized.

4.1 Need for Standardization Effort for Real-time Database Management

Industry and government have a need for real-time information management systems and in the absence of standards are developing proprietary products to meet real-time application needs. The proliferation of one of a kind solutions for real-time information management will only make interoperability harder later on. A real-time information management systems, is one which provides the information or response at a known time, but generally not before. Requirements for real-time information management are found in numerous military, credit card validation, medical, airlines and nuclear power systems. Presently there is a large movement in the DOD and industry to capture legacy systems and to incorpoate their information into on-line information infrastructures. Without real time database management systems standards in place as soon as possible, capturing legacy real-time systems databases may not be possible.

4.2 Existing Practice

This is a relatively new technology area. In the past (and in many existing systems) the data necessary for real-time applications (especially military) was hard coded into the system. Now, organizations are developing their own, proprietary solutions. Standards in this area would facilitate applications portability. New products being developed today and existing real-time database products are built upon existing real-time operating systems standard products such as POSIX 1003.4. As the market for proprietary, one of a kind solutions dwindles, real-time database vendors will realize the need and see the market for standard, commercial off-the-shelf real-time database systems. Organizations are unwilling to place large projects at risk of failure to support unique infrastructure services. The mandated trend is to use commercial off-the-shelf products to the maximum extent possible. This trend does not appear to be a short lived policy, rather a glimpse of what will become a common systems development policy.

4.3 Expected Stability With Respect To Current and Potential Technological Advance

Presently there is a trend in industry, academia and government to develop products and technologies that are based on open systems philosophies. A real-time operating systems standard, POSIX 1003.4, has been developed and is quickly being adopted into off the shelf products. The use of standards will enhance the ability of developers to build real time applications that will be portable across multiple platforms. Present practice of hand crafting real-time systems will be replaced with design philosophies that result in readily available commercial off-the-shelf open systems products. Presently real-time applications developers lack a consistent methodology for developing the information management portion of their application. Real-time database management systems standardization will result in products being developed that can operate on a variety of machines and provide a consistent platform for new applications developments. If the same open system philosophy of the operating systems domain is applied to the database management systems domain then current and future technological advances can be more readily supportrd within real-time database management systems products.

5 References

1. M. Aksit, J. Bosch, W. Van der Sterren and L. Bergmans. "Real-time Specification

Inheritance Anomilies and Real-time Filters." In *Proceedings of European Conference on Object Oriented Programming*, 1994.

2. R.~Abbott and H.~Garcia-Molina. "Scheduling real-time transactions." In *Proceedings of ACM SIGMOD*, March 1988.

3. ANSI."Portable operating systems interface standard." Technical Report ANSI, ANSI, Washington, DC., September 1993.

4. A. Biliris, S. Dar, N. Gehani, H. Jagadish, and K. Ramamrithham. "Asset: A system for supporting extended transactions." In *Proceedings of the ACM SIGMOD*, March 1994.

5. G. Bundell and G. Trivett. ``Real, real time transactions". *The Bulletin of the IEEE Technical Committee on Data Engineering*, 17(1), March 1994.

6. L. Cingiser DiPippo and V. Fay Wolfe. "Object-based semantic real-time concurrency control." In *Proceedings of the 14th Real-time Systems Symposium*, Dec 1993.

7. P. Fortier. "*D.Sc. Thesis: Early Commit.*" University of Massachusetts Lowell, 1993.

8. P. Fortier, J. Prichard, and V. Fay Wolfe. "Flexible Real-Time SQL Transactions." *Proceedings of the Real-Time Systems Symposium*, December 1994.

9. P. Fortier, D. Pitts, and T. Wilkes. "Experiences with data management in real-time C3 systems." *Proceedings of the SEDEMS II Conference*, April 1993.

10. P. Fortier and J. Prichard. "Concepts for a real-time structured database query language (RT-SQL)." *In the Proceedings of the IFIP/IFAC Workshop on Real-time Programming*, June 1994.

11. P. Fortier and J. Rumbut. " Issues and concepts for a real-time database management." *In the Proceedings of the First International Conference on Electronics and Information Management*, August 1994.

12. P. Fortier and CDR. G. Sawyer. "DISWG a new player in NGCR open systems standarads." to appear in *Computer Standards and Interfaces*, 1995.

13. P. Fortier and J. Sieg. "Recovery protocols for real-time database management systems." *In the Proceedings of the International Conference on Information Management (ICIM94)*, May 1994.

14. P. Fortier. "DBSSG, PRIS-TG: Database Management Systems Reference Model." PRISTG95-01, Febuary,1995.

15. L. Gallagher. "Object SQL: Language extentions for object data management." *International Society for Mini and Microcomputers CIKM-92*, 1992.

16. H. Garcia-Molina, D. Gawlick, J. Klein, K. Kleissner, and K. Salem. "Modeling long-running activities as nested sagas." *Bulletin of the IEEE Technical Committee on Data Engineering*, 14(1), March, 1991.

17. K. Gordon. "DISWG Database Management Systems Requirements". NGCR SPAWAR 331 2B2, Alexandria, Va., 1993.

18. M. Graham. "Real-time data management." *IEEE Technical Committee Real-Time Systems Newsletter*, 9(1/2), Spring/Summer 1993.

19. IITA Task Group. "Information Infrastructure Technology and Applications." National Coordination Office for HPCC, Executive Office of the President, February 1994.

20. D. Hughes. "ZIP-RTDBMS a real-time database management system." Technical Report PRISTG-93-011, ANSI DBSSG PRIS-TG, San Diego, CA, 1994.

21. ISO/IEC. "RDA - part 1: Generic model, service and protocol." Technical Report 9579-1, ISO/IEC, Washington, DC, 1993.

22. ISO/IEC. "RDA - part 2: SQL specification." Technical Report 9579-2, ISO/IEC, Washington, DC, 1993.

23. H. Korth, E. Levy, and A. Silberschatz. "A formal approach to recovery by compensating transactions." *In Proceedings of the 16th VLDB Conference*, 1990.

24. J. Melton and A. Simon. "Understanding the New SQL: A Complete Guide." Morgan Kauffman Publishers, San Mateo, CA., 1992.

25. J. Moss. "Nested Transactions: An Approach to Reliable Distributed Computing." PhD thesis, Massachusetts Institute of Technology, Cambridge, Mass., 1981.

26. H. Nakazato. "Issues on Sychronizing and Scheduling Tasks in Real-Time Database Systems." PhD thesis, University of Illinois at Urbana- Champaingn, Urbana, Il., 1993.

27. T. Ozsu and P. Valduriez. "Principles of Distributed Database Systems." Prentice Hall Inc., Englewood Cliffs, NJ, 1991.

28. J. Prichard, L. Dipippo, J. Peckham, and V. Wolfe. "RTSORAC: A real-time object oriented database model." *In: Proceedings of the 5th International Conference on Database and Expert Systems Applications,* Sept 1994.

29. K. Ramamritham. "Real-time databases." *International Journal of Distributed and Parallel Databases*, 1(2), 1993.

30. M. Roark. "RTDM: A real-time database management systems." Technical Report PRISTG-93-012, X3 DBSSG PRIS-TG, San Diego, CA, 1993.

31. L. Sha. *Ph.D Thesis: Modular Concurrency Control and Failure Recovery -- Consistency, Correctness and Optimality.* Carnegie-Mellon University, Pittsburg, PA, 1985.

32. L. Sha, J. Lehoczky, and E.D. Jensen. "Modular concurrency control and failure recovery." *IEEE Transactions on Computers*, 37(2), 1988.

33. S. Son, S. Yannopoulos, Y. Kim, and C. Iannacone. "Integration of a database system with real-time kernal for time-critical applications." *International Conference on Systems Integration*, June 1992.

34. J. Stankovic and W. Zhao. "On real-time transactions." *SIGMOD Record*, 17(1),

March 1988.

35. H. Tokuda. "Compensatable atomic actions in object-oriented operating systems." *Proceedings of the Pacific Computer Communications Symposium*, October 1985.

36. P. Watson. "The challenge of response time management in real-time distributed systems." In *IEEE Proceedings of the Fourth Israel Conference on Computer Systems and Software Engineering*, June 1989.

37. V. Wolfe, L. Cingiser, J. Peckham, and J. Prichard. "A model for real-time object-oriented databases." *IEEE Technical Committee on Real-Time Systems Newsletter*, 9(1/2), Spring/Summer 1993.

38. V. Wolfe, S. Davidson, and I. Lee. "Rtc: Language support for real-time concurrency." *Journal of Real-time Systems*, 5(1), March 1993.

39. P. Yu, K. Wu, K. Lin, and S. Son. "On real-time databases: Concurrency control and scheduling." In *Proceedings of the IEEE*, volume~82, January 1994.

STRATEGY AS A LEADING EDGE TO CREATIVE RELATIONAL DATABASE MANAGEMENT

Jan-Marie Esch

As database managers we face interesting challenges. Not only do we preform routine duties and maintain the system's integrity, we also have problems to solve. Many of the problems we face have to do with who needs what information, when, and in what format. Other problems have to do with who enters information, what needs to be tracked by the system, and how the entry of information affects what we can get out of the system and give those who need information. In this sense we work with users and management to make the system produce what is needed by the organization -- accurate and timely information. This session will help database managers sharpen their problem solving and team building skills using the following information:

I. Information needs assessments
   - A. Designing a needs assessment
   - B. Implementing a needs assessment
   - C. Analyzing the needs assessment

II. Involving staff and management in participative processes
   - A. Management involvement
   - B. Staff involvement

III. Creative implementations
   - A. Key factors to work with
     1. Tables
     2. Customizations

IV. Best-fit solutions
   - A. Giving feedback to team members (management & staff)

I. INFORMATION NEEDS ASSESSMENTS

The purpose of any needs assessment is to identify what is currently occurring, what needs to be occurring, and where the gaps are between what is and what needs to be. The information created by the needs assessment is the baseline information which allows for creative focus in solving the problems created by the gaps.

The information strategy of the organization is a central factor in the development of a relational database system. If a strategy has not been developed, now is the time to do it. The underlying factors to this strategy can be determined by a needs assessment; finding out who needs information, what information they need, when it is needed, and how the information needs to formatted. It sounds simple, but can be tricky. Managers are use to getting certain types of information in a certain format, and they may not be familiar with the potential of what new types of information they can get from a well managed system.

At least two assessments should be developed, one for management and supervisory level personnel who need information for making management/budget decisions, and a second one designed for users of the system who have to enter data and generate reports from the system.

Designing A Needs Assessment

The needs assessment covers four critical areas: who, what, when, and how. To design a good assessment, questions need to be developed and then answered by the people responsible for providing the guidance and decision making which is based on the data from the system. Since every organization is different in structure, size, composition, procedures and policies, every needs assessment will be somewhat different depending on the situation.

Questions need to be specific and measurable, that is, there must be some way of breaking the information gathered into meaningful groups or patterns. For a question to be measurable in some way, it needs "criteria" for measurement, the criteria will be different for each area of the needs assessment.

	Criteria	Questions
Who -		
What -		
When -		
How -		

Make sure that your questions use words and concepts which the interviewee will be familiar with. Most of the people you question will not be familiar with the intricacies of the system the way you are, and do not need to make decisions on how specifics are handled.

The groups or patterns of information will be identified during the analysis, and will help create a focus for problem solving activities. Once your questions are designed, test them out. You may want to ask the questions of someone who is not in a management or staff position, just to see how they answer. If you get unusual or unexpected responses, you may need to rework some of the questions before going forward with the implementation.

Implementing a Needs Assessment

How you carry out your needs assessment is absolutely critical to the success of your efforts. Making sure that you have identified the right people, how you go about including them will either make our break your process. All organizations have politics and agendas, work with these as much as you are able, don't buck the system too hard, but do cause enough ripple that it prepares the way for movement and change.

If you use an interview process, make sure that you:

1.) Be realistic about the time it will take to talk to the person and let them know how much time it will take.

2.) Keep the interviewee on track, don't allow yourself to get sidetracked on to other subjects for long periods of time.

3.) LISTEN. Be willing to accept criticism about what has been done with the system and information up to this point. This is a part of what is IMPORTANT, because you are trying to find out what is wrong so it can be set right. DON'T take the criticism personally, even though it may come across that way.

4.) Take LOTS of notes, or ask if you can record the interview so that you will be able compare data between people.

5.) Don't let what has "always been done" be a guide for what CAN be done in the future. Use follow-up questions to get more detail about what needs to be different.

6.) Share information. Let them know the purpose of what you are doing. Also use this as an opportunity to educate those needing what the system can provide about the potential for other types of information.

7.) Let the interviewee know this is part of a larger process that will take some TIME. If they expect immediate changes to their individual needs, they will be disappointed and your process will be undermined.

If you are using written questionnaires, many of the guidelines above need to be used in written format. What ever you do, with either type of process, try to be OBJECTIVE and OPEN to what you are hearing/reading. The purpose of this whole process is to make the system more effective in providing information, and to make it easier to use. Ultimately, this is to everyone's benefit, including yours.

Analyzing the Needs Assessment

Interview data may be harder to analyze, but you will have more direct feedback and can follow up with questions to get more detail. Direct contact with the managers and staff, combined with your openness, will also help develop relationships which will be of benefit during the problem solving phases of this process as well.

Advantages to written questionnaires include the ease in which the data can be collected and analyzed, but it is harder to get questionnaires back and does not help with the relationship process. You can use Likert scales, which are very easy to score; fill-in the blanks, multiple choice, or open-ended questions.

In any event, you will have to "sort" answers to questions from all your sources, for each one of the items/questions you asked. As you sort or score the answers, patterns will begin to emerge. Many departments may have problems with the same thing, or maybe only one has specific problems with one particular element of data. Go over the results very thoroughly and make sure that you develop an analysis that is CONSISTENT with what you have been told -- this will be critical in getting people more involved in a team process.

Your analysis should provide you with an overall picture of how the system is working/not working in providing information to people who need it. Once the sorting and scoring is done, pay close attention to GAPS in information and needs that are not being met. Some things may

be an easy "fix," while others may take more work, energy, and planning to create a good solution.

II. INVOLVING STAFF AND MANAGEMENT IN PARTICIPATIVE PROCESSES

The needs assessment is the first step in getting information and participation. It gives you an opportunity to find out what managers, supervisors, and staff may need from the system as far as information and reports are concerned. Your analysis of the needs assessment has given you information that you didn't have before, and you will be getting a clear picture of where the system is working as it should and where there are problems.

Management Involvement

Getting management participating is sometimes difficult due to busy schedules, it is sometimes accompanied by an attitude that what they are doing is somehow not related to what you (as a database manager) are doing. The needs assessment (if well done) will be the ice breaker for you. The next step in the process is giving feedback to management about your findings. Presenting your analysis to management level personnel in a group setting will give you an opportunity to share the information you have gathered. The suggestion of a group setting is particularly helpful if managers from different areas have conflicting needs from the system, this becomes an opportunity for them to see the diversity of needs you are dealing with as well as setting the stage for problem solving activities.

Follow-up meetings to work on specific problems should involve management that have conflicting needs, and others who might be affected by changes in strategy if something different (than what is currently being done) is implemented. Your purpose in holding these meetings is to facilitate the problem solving and then implement changes as needed. You are the one with the expertise who can educate managers on what can be done within the system -- they know what they need/want -- you have the power to make it happen (once you know).

Staff Involvement

The staff members using the system are very aware of their involvement with the system, and can become frustrated when the system doesn't do what they need. This is the "other" side of the system problem. Your analysis of the staff assessment should have given you a good outline of the rough spots which staff face. Again, some things may be easy fixes, while others may be more complex.

Hold follow-up meetings with staff to work on specific staff problems with the system. Some of the problems may be handled by additional training, or short-cuts within the system which will save them time and energy, which you can help them with. Don't be afraid to say, "I don't know, but I can find out," and then research what is needed.

Staffs' needs and managements' needs may relate to specific interrelated problems, or they may be separate. When the problems are interrelated, get the staff members involved in follow-up meetings with management to solve the problems together -- there is nothing worse than becoming a "go between" in a problem solving process that involves two groups of people -- getting them all into one place is the best way to get the "right fit" for the problem.

Your facilitation skills will be a key to successful management and staff involvement with the problem solving process. They are the team which will help define system strategy, but also give you the information you need to make the system successful in your organization. Remember to:

1.) Write key points down where everyone can see them.

2.) Define the problem clearly before starting to work on solutions.

3.) Add information and system know-how only when its needed to help solve a problem.

4.) When things seem stalled, summarize the major points of the conversation and then move the group forward through additional information or suggestions, or take a break for people to think about what has been presented.

5.) Don't be afraid to draw diagrams, pictures, or use other methods to get the problem defined or to share information that is more difficult to understand. Keep "systemeese" out of your presentation and information -- keep things simple and easy to understand.

6.) Problem solving always takes more time than expected, don't lead team members to believe that they will be able to solve all the problems at once -- the system and strategy have to develop over time (most creative endeavors are this way).

7.) Do research between meetings so that you are prepared with additional information which will help with the problem solving.

Guiding the process is more effective than directing, you will get more information and solve more problems using facilitation skills.

II. CREATIVE IMPLEMENTATIONS

Once the problem solving is done the implementation is in your hands. The technical part of the implementation is what is going to take you the most time. If you have done the research and facilitated the meetings, you have a pretty good idea what you need to do next.

Key Factors

As database managers we have access to knowledge and skills that will allow us to fix system problems, which others in our organization don't have. A through knowledge of your system is important, keeping track of customizations, documentation of why customizations and fields are used in particular ways is equally important. The main tools you have for making the system work they way it needs to for your organization are the tables and developing customizations.

Tables. Developing strategies for the use of tables is one of the best tools you have. You can get very creative with tables. Knowing that you can sort by any field in the system provides you with the opportunity to use fields in consistent and useful ways.

Some considerations in developing tables and strategies for fields:

1.) Does this field/table get used for sending information to other systems, and what do the other systems require? This may help determine codes within the table or field.

2.) Does more than one type of sorting activity occur off of a single field? This will help determine the length and structure of the field.

3.) Does the information appear in reports? This may help determine abbreviations, since they need to be easily understood.

4.) Keep the codes reasonable, a 6 space code could designate 3 elements (for example: MKAS02 - Medical Kaiser Two Party insurance).

5.) Keep codes of consistent length within a table, which allows for easy reading.

6.) Don't be afraid to be creative with codes, they are flexible which is part of what makes them so useful.

Customizations. Customizations are your second tool. They are very powerful, and can radically change what kinds of information you can provide. It can be as simple as creating a new field, or attaching a field to an already existing table, or removing or adding fields to different screens.

Here are some considerations for customizations:

1.) Is there a field that already exists that will sort in the way it needs to be sorted? Don't create headaches by building new fields you may not need, it may be as simple as writing a definition which will convert that information into a different output for a specific report or data transfer.

2.) If it is a major change, is there a module or program which can be incorporated to do what you want without a lot of customization and individualized programming? If it would cost more to create it yourself, you may be making more work you don't need.

3.) Will the customization impact any other functions or displays, which may cause problems for users doing data entry? Adding a new field to a screen can change the input sequence, check with users to see where they would like the field to appear.

4.) If it is a major change, duplicate the system elsewhere and work on the duplicate before implementing it on the live system. This allows you to test the changes and makes sure that you don't lose any data.

5.) Keep clear and concise notes on all customizations and the reasoning behind them. A record of what you have done will help you in the future, because you will have all the strategy and reasoning behind the structure of the system.

6.) Duplicate and play. If you are thinking of changing a definition so that it will do something different than originally designed to do, make a synonym and program

the changes into it -- then test it out first before using the new definition. Again, keep FASTIDIOUS notes on these changes -- if it doesn't work delete it.

7.) Don't be afraid to get help -- user groups and support lines are for this purpose. They may not know your strategy or understand what the purpose of the change is, but they may have technical tips which will make it easier for you.

IV. Best-Fit Solutions

You have used the best minds (the ones needing the information and help) in your organization to help develop the system. This is what will help make the best-fit. If we try to dream up all the answers ourselves, we may be wrong. Getting people involved and giving them what they need builds the type of team atmosphere needed to be successful as an organization, as well as building support for the system and its many uses. Once you have begun this process, it becomes a natural part of the database management process and provides many creative opportunities.

Giving Feedback to Team Members

Make sure that you keep in touch with your team members, let them know about progress toward goals, when they will begin seeing changes, and also how much you appreciate their help in the process. The goal is to make the system work for the organization and provide the information and help that people need. Good luck with your creative process!

BIOGRAPHY

Jan-Marie Esch is an organization development specialist, currently working with information needs and relational database development, at County Sanitation Districts of Orange County. She holds a B.A. in Education, an M.S. in Organization Development, and is currently completing a Ph.D. in Leadership and Human Behavior. Managing and programming in Revelation based relational databases, she has used her organization development skills and educational abilities to build information teams at the Districts. Her innovative ideas and facilitation skills have combined to support organizational members at all levels.

County Sanitation Districts of Orange County
Human Resources Office
10844 Ellis Avenue
Fountain Valley, CA 92728

(714) 962-2411 x2103; FAX (714) 962-0427

USING EXPERT SYSTEM TECHNOLOGY TO STANDARDIZE DATA ELEMENTS

JENNIFER LITTLE, AMERIND, INC.

1. INTRODUCTION

Data are symbols that represent some thing. Human perception of data provides meaning and results in information. Putting information in a broader context that humans use to determine significance yields knowledge. Applying knowledge in anticipation of events to make a valuable difference is wisdom. Therefore, data are the foundation for information, knowledge, and wisdom.

Expert systems can capture the essence of the expertise locked up inside a precious few individual analysts and be used to leverage that expertise by accomplishing more, accomplishing it faster, accomplishing it at a lower cost, and/or accomplishing it more consistently. The data element standardization process could be captured by an expert system and it could be used to standardize more data elements, standardize them faster, and reduce the cost of doing it.

Note, the examples used in this article were drawn from the Department of Defense because it is believed that in establishing their data administration functions, they have performed exhaustive research and have selected the best available components (e.g., definitions, rules, policies, concepts).

2. WHY AN EXPERT SYSTEM?

Most data element standardization programs use an automated tool to assist the process. Data modeling tools, data dictionary tools, repository tools, off-the-shelf tools, and in-house developed tools are used. Since these tools were never designed to tackle the specific problem of data element standardization, they have inherent limitations to their ability to adequately support the standardization process. That is not to say that they have not been helpful in standardizing data elements, because they have been extremely helpful, and the progress made thus far could not have been possible without them. However, it has been due to the expertise of the human analysts in linking, contorting, or forcing the tools to give them what they needed to perform the bulk of the analysis themselves. Several data element standardization efforts probably have kernels of expert systems built already because that is what they needed to do their jobs.

An expert system will improve the standardization process in several ways. It will make the process faster, produce higher quality products, and be cheaper. An expert system will accelerate the standardization process by performing repetitive processes rapidly. As mentioned above, several data element standardization efforts probably have already enhanced their tools to do part of this. However, they may be, from working in isolation from other groups, be producing inconsistent results. The process of an individual standardizing one data element can be accelerated using an expert system, and the overall process for an organization can also be

accelerated by using a consistent support tool. The overall quality of the data elements produced will increase by using an expert system because all the data elements will be produced by the "expert" rather than just the ones coming from the project groups lucky enough to have an in-house expert. The overall cost of standardizing data elements will reduce dramatically by using an expert system because it will no longer take analysts years to become experts at standardizing data elements, nor will the work need to be contracted out. An expert system will also be able to assist in the complex semantic analysis that is required by standardization programs, but for which guidance is only addressed indirectly.

3. DATA ELEMENT STANDARDIZATION AS THE PROBLEM DOMAIN

What is data? What is a data element? What is a standard data element? Each organization that wants to standardize their data elements must answer these questions. It would be redundant to redefine these terms, therefore, figure 1 contains an example of the Department of Defense's definitions of them.

Data. A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by humans or by automatic means.
Data Element. A named identifier of each of the entities and their attributes that are represented in a database.
Data Element Standardization. The process of documenting, reviewing and approving unique names, definitions, characteristics and representations of data elements according to established procedures and conventions.
Standard Data Element. A data element which has been submitted formally for standardization in accordance with the organization's data element standardization procedures. [1]

Figure 1

Purpose of Standardizing Data Elements

Standard data elements aid many data information management objectives. One of the more obvious objectives of information management and data administration is ensuring the quality of the data. What is quality with respect to data? "Determining quality is meeting *customer requirements and expectations*." [italics theirs] [2] and a standard data element is the specification of the customers' data requirements. Stated another way, since standard data elements define customers' requirements, they specify the quality standards to be used in measuring the quality of that data. When the quality of data can be measured against the standard data element, it can be reported, and improved when necessary. When the data meets all the requirements, then it is the highest quality data. To the extent it does not meet the requirements, its quality is lower. The quality can then be measured and be given a quantitative representation.

Organizations standardize their data because they want higher quality data and data that are reusable. Standard data elements also provide an accurate and consistent understanding of the data to users. In this respect the data element provides the accepted definition of the data. Lastly, standard data elements provide information system developers with consistent raw materials from which to build databases more quickly and easily.

The Data Element Standardization Process

The data element standardization process is a complex one. It is partly influenced by the technology used to implement automated information systems (e.g., database management systems (DBMS) employing the relational model, and flat files); the semantic differences among speciality management functions/disciplines (e.g., finance, inventory, and personnel); and the political power held by those different specialty functions/disciplines. For an organization to be successful in standardizing their data elements, experts are required.

The process of standardizing data elements is one for which few experts exist. To make matters worse, the expertise required is of two kinds: (1) technical expertise, which is knowledge and skill in how to standardize data elements well, also referred to as the syntactical characteristics; and (2) functional expertise, which is knowledge of the content and context of the data elements, also referred to as the semantic characteristics. Data element standardization criteria typically addresses the syntactical characteristics well, but it can only address the semantic characteristics indirectly because the content and context of each data element will be different.

What makes a data element a standard data element? Two things: (1) compliance with a set of standardization rules, and (2) approval within the organization's standardization program. No two organizations' rules are exactly alike. However, they all include some general principles, such as, standard data elements must be unique, defined and named well.

A part of the standardization policies and procedures that exist for the Department of Defense is shown in figures 2, 3, and 4 as an example. These policies and procedures contain many requirements for data element design, data element definition, and data element naming. These requirements could be implemented easily in an expert system, but are extremely difficult to implement consistently without one. For example, a core requirement for standard data elements included in figure 2 (underlined) is that they "must not have more than one meaning. A data element should reflect a single concept to promote shareability and data independence from applications using the data element." This is a critical requirement for data elements, and it is a very complex requirement to fulfill. Can this be done by looking at the data element? Are there things to look for? Should the data element be compared with anything else to determine if it includes only one concept? The answer to each is yes. The human data element standardization experts have developed heuristics for performing this analysis, and their results are fairly good. However, their results are not always consistent and they take a long time to achieve because of the scarcity of those experts. (See "One Concept Equals One Data Element: A Rule for Developing Data Elements" for more on this topic.) [3]

The quality of the data element is the key to the sound foundation for all data structures. Proper emphasis on the creation, naming, and definition of data elements will improve the quality of the entire data structure. Standard data elements should be based upon the data entities and data entity attributes identified in the DoD data model, or recommended for expansion of the DoD data model from a lower level data model, to ensure maximum shareability and interoperability of data throughout the Department of Defense. Several considerations are important to the quality of the data element.

1. Data elements must be designed:
   a. To represent the attributes (characteristics) of data entities identified in data models. A model-driven approach to data standards provides a logical basis for, and lends integrity to, standard data elements.
   b. According to functional requirements and logical, and not physical, characteristics. Physical characteristics include any connotations regarding technology (hardware or software), physical location (databases, records, files, or tables), organization (data steward), or application (systems, applications, or programs).
   c. According to the purpose or function of the data element rather than how, where, and when the data element is used or who uses it. It indicates what the data element represents and ensures common understanding.
   d. So that it has singularity of purpose. Data elements must not have more than one meaning. A data element should reflect a single concept to promote shareability and data independence from applications using the data element.
   e. With generic element values (domain) that are mutually exclusive and totally exhaustive when the class word 'Code" is used.

2. Data elements should not be designed with:
   a. Values (domain) that may be confused with another value in the same domain. For example mixing similar numbers and letters such as: 0/O, 1/I, 2/Z, U/V and 5/S.
   b. Values (domain) that have embedded meaning or intelligence within part of the code when the class word 'Code' is used. For example, do not develop a multiple-character code where in the value of one or more of the characters in the code have special meaning (i.e., a benefits plan code such as "201," "202," "204," or "205," where the last digit identifies a particular option within the benefit plan).
   c. Overlap or redundancy among the purpose or use of different data elements (e.g., "Birth Date," "Current Date," and "Age"). [5]

Figure 2

C. DATA ELEMENT DEFINITION

The definition and naming of a data element is an iterative design process with the data element definition often being modified as the data element is being developed.

1. Data element definitions must:
   a. Be based on the definitions of data entity attributes established in the DoD data model or established in an approved data model linked (mapped) to the DoD data model.
   b. Have a structure which centers around the generic element of the data it describes. Developing a standard data definition using a structure minimizes "writer's block" and facilitates the development of consistent and meaningful definitions that can be accepted by all users. Examples of data definition structures for each class word are contained in Appendix A, below.
   c. Define WHAT the data is, not HOW, WHERE, or WHEN data are used or WHO uses the data.
   d. Be more than just a reiteration of the data element name. The definition must add meaning to the name and not merely rephrase the name. The class word is an exception, its meaning does not need to be redefined in each definition.
   e. Describe its purpose and usefulness and must not contain physical characteristics. The definition must describe logical, not physical, qualities.
   f. Have one and only one interpretation and must not be ambiguous. Terms with differing or varying connotations must have their meanings clearly explained in the definition.

2. Data element definitions must not:
   a. Contain conjunctions or phrases indicating multiplicity of purpose of a data element, ambiguity of definition, or process orientation.
   b. Contain technical jargon that may be unfamiliar to the reader.
   c. Contain acronyms and abbreviations.
   d. Restate the characteristics of the data element. For example, do not use statements or phrases such as "...seven characters in length..."or"... an alpha-numeric code..." in the definition.
   e. Restate a process definition that describes how a data element is calculated, derived, assimilated, or manipulated.
   f. Contain information about the valid values or domain of the data elements.
   g. Be circular. A situation cannot exist where one definition points to a second definition for further explanation and the second definition points back to the original definition. [6]

Figure 3

D. DATA ELEMENT NAMING

The set of guidelines for naming data elements establishes a naming convention, or classification scheme, that will make it easier to determine if a data requirement is already being met within the Department of Defense or if it is a new requirement that needs to be fully defined and the data collected and distributed as necessary.

1. The names of data elements should:
   a. Be based on the names of data entity attributes identified in the DoD data model or an approved data model linked (mapped) to the DoD data model.
   b. Be clear, accurate, and self-explanatory.
   c. Be named according to logical, and not physical considerations. Physical characteristics include any connotations regarding technology (hardware or software), physical location (databases, files, or tables), organization (data steward), or function (systems, applications, or programs).
   d. Consist of the minimum number of words that categorize the data element. Fewer words may be too general while more words may be too narrow or restrictive. Modifiers may be used with class words, generic elements, and prime words to fully describe generic elements and data elements. Modifiers are often derived from the data entity attribute names and the entity names identified in the DoD data model or an approved data model linked (mapped) to the DoD data model.
   e. Include only alphabetic characters (A-Z, a-z), hyphens (-), and spaces().
   f. Have each component of the name separated by a space.
   g. Have multiple word prime words connected with hyphens. Examples of multiple prime words might be "Purchase-Order," "Medical-Facility," or "Civilian Government."

2. The following are not permitted in data element names:
   a. Words which redefine the data element or contain information that more correctly belongs in the definition.
   b. Class words used as modifiers or prime words.
   c. Abbreviations or acronyms. (Exceptions to this rule may be granted by the DoD DAd in the case of universally accepted abbreviations or acronyms. The DDRS will contain a list of approved abbreviations and acronyms.)
   d. Names of organizations, computer or information systems, directives, forms, screens, or reports. e. Titles of blocks, rows, or columns of screens, reports, forms, or listings.
   f. Expression of multiple concepts, either implicitly or explicitly.
   g. Plurals of words.
   h. The possessive forms of a word, i.e., a word which denotes ownership.
   i. Articles (e.g., a, an, the).
   j. Conjunctions (e.g., and, or, but).
   k. Verbs.
   i. Prepositions (e.g., at, by, for, from, in, of, to).[7]

Figure 4

Difficulties in Standardizing Data Elements

Many organizations that are standardizing their data elements are having difficulties. This is true of organizations that have had standardization programs in place for several years as well as organizations that have recently begun their standardization efforts. The difficulties organizations are having can be grouped into three common categories:

Time related - Example: An organization does not have the data elements standardized in time to use them in a new information system being developed so the system is developed without them and plans are added to retrofit to the standards later.

Quality related - Example: An organization expends resources to standardize their data elements in their accounting system and is appalled with the resulting accounting specific data elements because they expected the data elements to reflect their entire organization's understanding of their financial data.

Cost related - Example: An organization hires several analysts to standardize their data elements during a multi-year project, but the resources are diverted in the middle of the project and the standardization remains unfinished and unusable because it was done with a monolithic perspective.

There are likely many causes of these difficulties that are unrelated to the data element standardization process, such as changes in organization strategies, changes in the marketplace, etc. However, some of the difficulties may be caused by characteristics of the standardization process. For example, the standardization criteria may be conflicting or overly stringent. Data administrators have been accused, at times, of being perfectionists and of sacrificing practical results for the perfect data element. Some organizations have eased their standardization requirements after initial attempts to implement them resulted in resistance from the people responsible for implementing the standardization criteria. Another characteristic of the standardization process that may cause difficulties is relying on inexperienced personnel that lack adequate training. Lastly, there may be a lack of adequate tools to support the process. Some of the tasks to be performed as part of the standardization process cannot possibly be performed by humans alone, and they were not intended to be performed by humans alone (e.g., searching numerous existing data elements for similarities and linking data elements to attributes in data models).

Choosing Expert System Technology

There are other characteristics of a task that need analysis to determine whether applying expert system technology is appropriate. The following list is from Waterman's work on expert systems. [4]

- Is symbol manipulation required? - Most problem solving tasks require symbol manipulation (the exceptions are usually mathematical problems). Symbol

manipulation is central to the process of standardizing data elements because data are symbols that represent complex meanings. Using data and data elements instead of uncontrolled text or prose, forces reliance on a limited set of symbols to convey complex meanings.

- Are heuristic solutions required? - As discussed above, the human experts at standardizing data elements have become experts based on their ability to develop and to use rules of thumb(heuristics) to guide them in constructing valuable data elements.

- What is the complexity of the task? - It should be obvious that, if the task of data element standardization were easy there would be few problems with the process; and therefore, an expert system would be unnecessary. On the other hand, expert system technology is not suitable for a problem that demands days of computer processing to analyze and present solution options. Standardizing data elements is a task that while not beyond human capabilities to perform, would be substantially improved by applying expert system technology.

- What is the practical value of the task? - Expert systems provide solutions to be applied to a functional need, not just exploratory analysis of theoretical issues. Because standard data elements have been recognized as valuable organizational commodities, developing better data elements and developing them more quickly and consistently is a pragmatic goal.

- What is the size of the task? - The task of standardizing all the data elements of an organization is a large (and often daunting) task. However, standardizing one data element at a time is a task performed by individuals. Although teams of people usually work on sets of data elements at a time, several people are not required to all work on one data element in order to standardize it.

4. APPLYING EXPERT SYSTEM TECHNOLOGY

What Is an Expert System?

"A computer program that uses expert knowledge to attain high levels of performance in a narrow problem area. These programs typically represent knowledge symbolically, examine and explain their reasoning processes, and address problem areas that require years of special training, [experience] and education for humans to master." [8]

"As originally used, a computer system that could perform at, or near, the level of a human expert. The popular press and various software entrepreneurs have already used the term in so many ways, however, that it now defies any precise meaning." [9]

Expert systems like many other components in the fast-paced world of technology do not have a definition that all the experts agree on, as the examples above indicate. However, even without an accepted and precise definition, expert systems can be grouped by common characteristics. Harmon, et. al., group expert systems into six major types : (1) procedural, (2) diagnostic, (3) monitoring, (4) configuration/design, (5) scheduling, and (6) planning. The characteristics of diagnostic expert systems most closely describe the type of expert system needed for the data element standardization process. As the name implies, a diagnostic expert system would be used to diagnose the situation and provide advice for the next course of action. It is the most common type among expert systems developed.[10]

Roles Expert Systems Serve

Another way to group expert systems is by the role they serve when used by humans. Expert systems can be used in a variety of roles. The type of problem domain needs to be taken into account when choosing the role of the expert system, and the role in which the expert system will serve must be decided before the design process begins. The relationship between the users and the system also strongly influences the choice of the role for the expert systems. An expert system must be accepted by the users in the role that it is developed to fill; otherwise, it will be ineffective. Lastly, the resources available to develop the expert system must be taken into account when selecting the role of an expert system. The more technically sophisticated roles demand more technically sophisticated development processes. The following discussion of roles is summarized from K. Pedersen's book on expert systems.[11]

The least demanding role an expert system can play is as an advisor. This is the role that a data element standardization expert system would fill best. In this role, the expert system helps ensure consistency among many analysts performing the same task. The advising role also helps train analysts that have general experience with the functional area (i.e., data administration, data management, or information resource management), but who are not skilled in implementing the specific detailed requirements. The expert system that performs in this role is not expected to perform on par with the human expert's that perform this task; it is expected to provide guidance and act as a "sounding board." Advisor expert systems are well suited for domains that demand human decision making, such as those that address potential life threatening situations or that put large amounts of valuable resources at risk.

The other two roles an expert system may fill require a higher degree of trust in the system for it to be productive. In the peer role, the expert system performs as an equal to the human analysts. The analysts fully investigate the suggestions that the expert system provides, even when the suggestions differ from their own. The expert system provides the reasoning it has applied to explain why its conclusions may be different from those of the human analysts. The most technically sophisticated role for an expert system is an the expert. In this role, the system is used by analysts who have minimal experience performing the task. The expert system, in this role, is the decision maker.

Developing the Expert System

Once the choices of type and the role have been made, the systems development process can begin. This process has some similarities to the development processes for more traditional types of information systems and some differences.

Specifying the requirements and identifying the problem and scope, the participants, and the goals and objectives of the system are done during the initial phase of the development process similarly to the more traditional systems development processes. So are the tasks of identifying the technical environment and developing technical configuration constraints. The result of this initial phase is a specification of the requirements. The selection of participants is slightly different from traditional systems development in that the human experts at the current task need to be involved so their expertise can be captured, whereas traditional systems development tends to involve representative users of the proposed system.

The next phase is only similar to traditional systems development in that it may be likened to selecting the programming language at a gross level of abstraction. This phase includes choosing the basis for the problem solving strategy to be used by the expert system. Fortunately, this issue is easier to address since the appearance of expert system shells. Expert system shells are empty expert systems that only include the basis for solving the problem. Ruled-based is a common type of expert system shell. Rule-based expert system shells are set up to have rule sets or knowledge bases loaded into them that will be used to perform the expert analysis. These rules are not simply a series of consecutively preformed procedures. They are heuristics that are applied to the problems presented for solution by users and are merged and the overlaps and the possible contradictions reconciled by the expert system in order to provide the expert advice to the user. There are also domain specific, inductive, and hybrid types of expert system shells. Choosing an expert system shell will focus the rest of the development tasks, but it may also limit the flexibility of the solutions to be provided. The limitations caused by selecting a rule-based expert system shell for developing a data element expert system are neutralized by the choice of the advisor role for the data element expert system, which is the least demanding role for an expert system and because the data element standardization problem domain is well documented and has many rules itself.

In the phase that follows, explaining the process that the human experts go through in terms that can be: (1) verified by the human experts as accurately reflecting what they do, and (2) translated into components the expert system shell can understand is the main activity. It might not be possible to use one diagraming and documentation technique to achieve both goals of this explanation process. Additionally, a model of the process the humans use may already exist. In either case a translation between two or more models may be necessary to accomplish both goals of this phase. This phase will include many detailed conversations with the data element standardization experts asking "how do you do that?" and asking "how do you do that?" in response to the first question, and continuing in this manner until the essential rules of thumb are uncovered.

The next phase results in a working system. The rules uncovered and described in a way that the expert system shell can comprehend are loaded into its knowledge base and testing begins. Data elements that have already been standardized by the human experts can be used to test the advice provided by the expert system. When the results do not match, the differences will point to the rules that were not discovered during the previous phase. These "new" rules need to be documented as the others were and added to the knowledge base. This version of the expert system can be thought of as a prototype in that it only implements the performance of a small group of the human experts. It may need to be expanded to reflect a larger set of situations and relevant rules or it may be appropriate that it reflects only the small group of human experts. They may be the only real experts that exist, and the purpose of the system may be to assist the rest of the humans performing the task to produce consistent results.

5. CONCLUSION

Expert system technologies have been available for a number of years. Organizations have been standardizing data elements for a number of years. Why are there no commercially available data element standardization expert systems? It could be part of the shoemaker's children syndrome. The fact that there are some tools that partly address the problem could be part of the answer. Maybe commercial software developers are not aware of the need for this technology.

A more important question may be, what will it take to create a data element standardization expert system? I suggest that a collaborative partnership is needed. A recommended consortium consisting of: (1) a research institution (government, university, etc.), (2) a client (government or commercial company with a need for standard data elements and the functional area knowledge), (3) a technology expert (government or commercial software developer with expert systems/artificial intelligence expertise), and (4) some data element standardization experts (again, government or a commercial consultant firm with this expertise) could jointly tackle the problem and jointly benefit.

6. REFERENCES

[1] Office of the Assistant Secretary of Defense (Command, Control, Communications, and Intelligence), Department of Defense Data Element Standardization Procedures, Department of Defense, Washington, DC, 1993.

[2] Federal Quality Institute, Federal Total Quality Management Handbook, United States Office of Personnel Management, Washington, DC, 1991.

[3] J. Little, One Concept Equals One Data Element: A Rule for Developing Data Elements, Auerbach Publications, New York, 1994.

[4] D. A. Waterman, A Guide to Expert Systems, Addison-Wesley Publishing Company, Reading, MA, 1986, p.132.

[5] Office of the Assistant Secretary of Defense (Command, Control, Communications, and Intelligence), Department of Defense Data Element Standardization Procedures, Department of Defense, Washington, DC, 1993.

[6] ibid.

[7] ibid.

[8] D. A. Waterman, A Guide to Expert Systems, Addison-Wesley Publishing Company, Reading, MA, 1986, p.390.

[9] P. Harmon, R. Maus, W. Morrissey, Expert Systems Tools and Applications, John Wiley & Sons, New York, 1988, p. 265.

[10] ibid, pp. 51-53.

[11] K. Pedersen, Expert Systems Programming: Practical Techniques for Rule-Based Systems, John Wiley & Sons, New York, 1989, pp. 35 - 36.

Jennifer Little is a Program Manager for AmerInd, Inc. in Alexandria, VA. She is also serving her second term as the President of the Data Administration Management Association, National Capital Region (DAMA-NCR). Ms Little has concentrated on integrating data administration techniques with other technical and management disciplines, such as systems development and business process reengineering. She has been supporting data management programs at several federal agencies for the last ten years.

Standardized Metadata and Metadata Capture Tools for Migration to New Operational Systems and Development of DSS Infrastructures

by

Duane Hufford
American Management Systems

Legacy system environments that exist today have evolved through years of development and maintenance activities responding to the ever changing organization and functions of the enterprise. Although data represent the most staple component of the information system environment, large organizations are currently faced with massively disparate data when trying to standardize their representations across multiple jurisdictions and disciplines. Metadata provide a very important source of documentation for bringing order out of this confusion.

This paper describes metadata and metadata capture techniques that are important for two very active integration practices in commercial industry and the Department of Defense:

- Reverse engineering legacy transaction systems to open systems environments consisting of standardized data in enterprise databases.
- Developing decision support architectures that extract snapshots of data from the transaction system databases for new uses in DSS applications.

When databases are installed to support transaction systems or DSS applications, user oriented metadata (e.g., documentation describing the database) are seldom included. Omitting this metadata leaves users spanning different jurisdictions and disciplines with no accessible source of information for understanding the abbreviations used by the database beyond what may be provided on system data entry and help screens. Furthermore, these metadata are critical for facilitating reuse of system assets and improvement of data integration under future system development/maintenance efforts.

To demonstrate how metadata can be used to bring order out of the chaos currently reigning over the data of many organizations, this paper:

- Describes a case example of how metadata on data element domains was critical for reverse engineering an Army Budget System during its migration to a client server environment. Techniques used for capturing this metadata and developing the data model will be summarized and contrasted with 'normal' practices for top-down data modeling.
- Summarizes metadata requirements shared by the example reverse engineering effort and the DoD Data Element Standardization Program.
- Identifies tools available on the marketplace that capture and put metadata to practical use for the data warehouse administrators and end-users in the DSS environment. The paper will identify types of metadata required by these different tools and the expanding role metadata plays throughout an open DSS architecture.

- Describes commonly cited requirement short-falls of the current suit of tools supporting DSS architectures, and shows how many of these problems relate to a common set of metadata documentation problems faced by data administrators and data modelers.

- Provides recommendations for modifying data administration practices based on emerging metadata requirements of commercial tools and the short-falls observed.

A. Domain Analysis for Legacy Applications

Many of the data modeling tools available today focus on the entity-attribute-relationship paradigm for modeling data. They offer validation facilities to detect errors in the syntax of a data model at the entity-attribute-relationship level and often provide facilities to ensure that the data model's syntax is correct. But in order to make sure that the data model is correct, these tools need to provide more support for domain analysis (i.e., research into the specification of allowable values). To illustrate this point, consider a data modeling effort for the US Army's Program Analysis and Execution (PA&E) Directorate to help 'rightsize' the Program Budget and Execution (Probe) system.

The objective of the Probe Rightsizing effort was to redeploy the Probe system in a client/server architecture that would allow better investigation of probe data during program planning and result in more accurate budget preparation. This project involved reverse engineering the data model following procedures graphically depicted in Figure 1

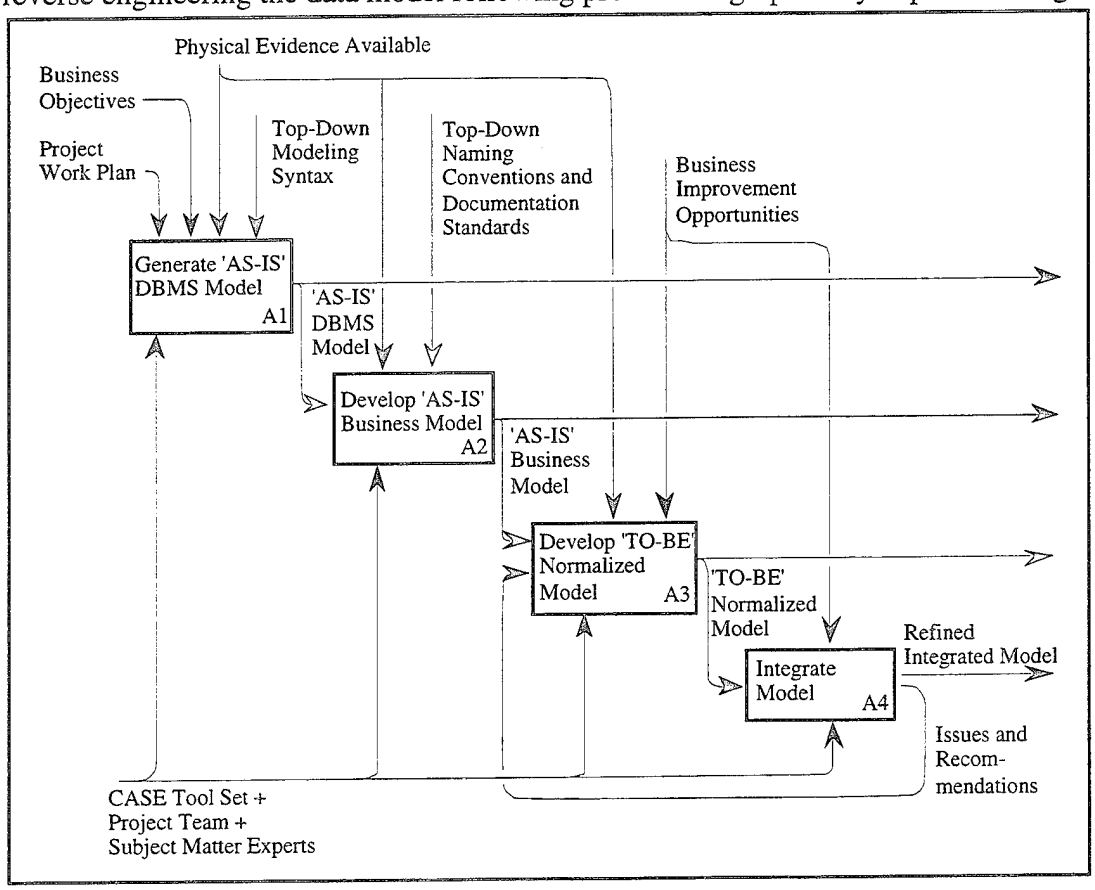


Figure 1: Data Reverse Engineering Procedures Used on the Probe Rightsizing Project

to fully specify all the business rules that needed to be enforced by the system. Reverse engineering was appropriate because the size of the existing database was reasonably sized for the project team (i.e., approximately 50 tables), and the data requirements for the new system were largely represented in the database for the existing system.

The procedures identified in Figure 1 are described in the sections that follow, highlighting the benefits of the domain level analysis to the project team, and contrasting the approach with normal practices for top-down data modeling.

A.1 Generate 'AS-IS' DBMS Model

The data modeling team conducted an archaeological dig into the legacy Probe database and synthesized a DBMS strawman model from the structure of keys and a quick reconnaissance of the existing application. The 'AS-IS' DBMS model was documented using IDEF1X syntax, but no effort was made to change the names of columns and tables. A portion of the Probe DBMS model shown in Figure 2 illustrates how the DBMS model looked. A purely 'top-down' approach to data modeling is denied the benefits of this 'strawman' data model, and many of the iterative validation techniques that will be highlighted below. Using a top-down data modeling syntax such as IDEF1X to represent the DBMS model helped the project team identify components of

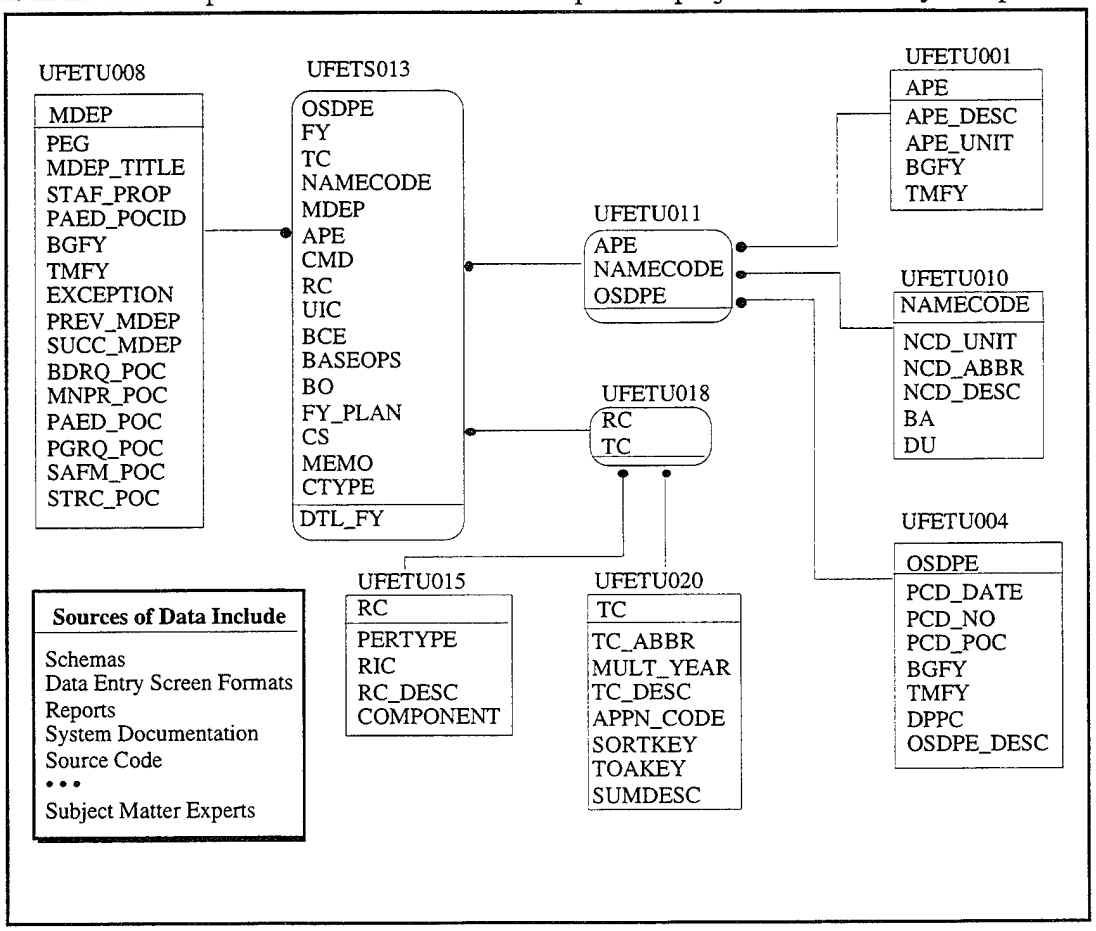


Figure 2: DBMS Model -- Transcribed from the Database and Fit to a Data Modeling Syntax

the model most likely to be hiding complex business rules that would require concentrated research. For example, the table labelled as 'UFETS013' in Figure 2 contains 16 data elements in its primary key. This is particularly intriguing because the counts for elements in the primary keys of related tables are substantially smaller (e.g., at most three data elements).

Although we can often detect that something is hidden and complex by inspecting a DBMS model, it takes additional domain analysis to determine and document exactly what the problem is. To support downstream analysis required to uncover the hidden business rules and improve the model's documentation, the content for 'AS IS' Probe database was copied into the client/server development environment where it could be easily accessed and queried using SQL.

A.2 Develop 'AS-IS' Business Model

The data modelers collaborated closely with subject matter experts in focused data modeling work sessions to:

- Develop full names and definitions for each data element and table appearing in the reverse engineered DBMS model.
- Identify that nature of any relationships between tables and data elements appearing in the DBMS model.
- Identify sources of information for validating each data element's values both when considered individually and when correlated with the values of other related data elements.
- Identify and document any new data requirements.

Figure 3 illustrates the nature of the re-naming that took place and the type of documentation gathered during these sessions. The domain level documentation was critical for following through on discussions with the subject matter experts. The data modeling team made a practice of verifying their understanding of what the subject matter experts said by running SQL queries designed to test whether there were any violations to these business rules. Frequently violations were detected and when the rationale for these violations were resolved with the subject matter experts, three common reasons emerged:

- The data values were in fact wrong. The PA&E staff was always happy when these errors were detected because the budget programming staff could then fix the errors. Over eleven thousand data values errors were detected in the course of this project through this type of domain analysis.
- The data modelers misunderstood the subject matter expert or the subject matter expert misunderstood the nature of the question as originally posed by the data modeling team. When this happened the data modelers were always happy that they found out about the error early. Domain analysis was critical for bottom-up quality assurance throughout this effort.

- There were additional business rules related to other elements that had been missed in the earlier discussions. Domain analysis helped expose gaps in the information provided to the data modeling team.

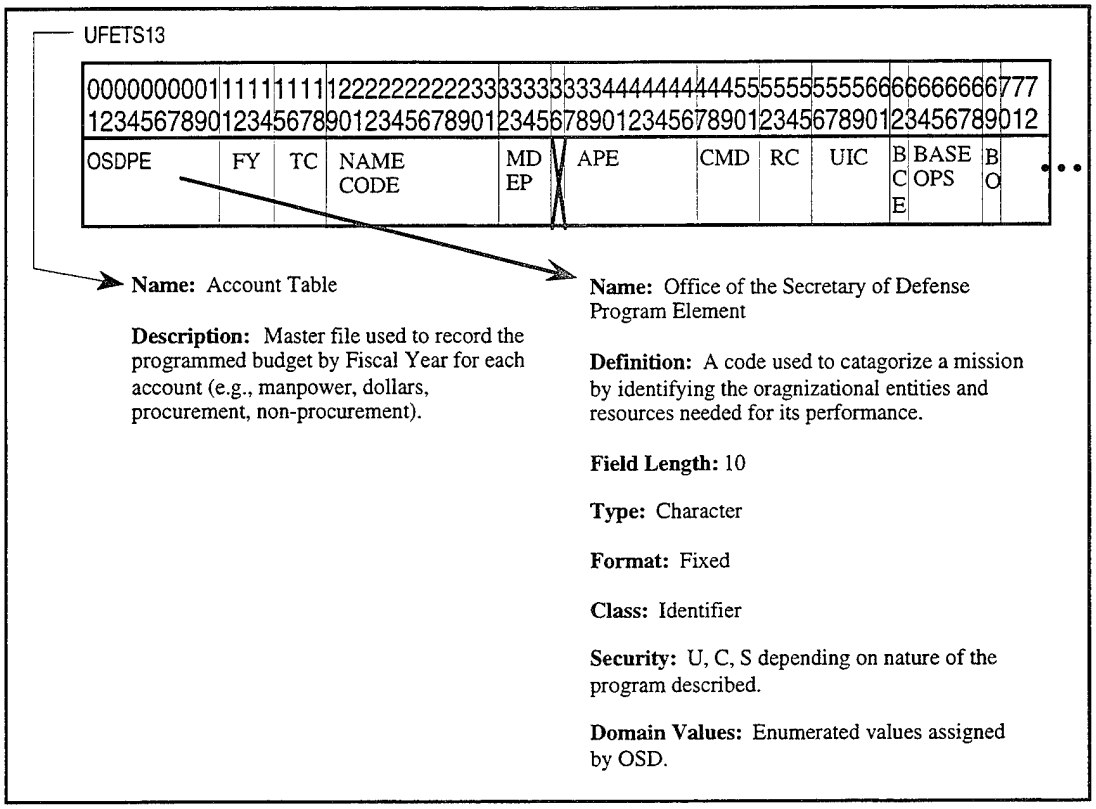


Figure 3: Example Mapping of Physical Evidence to Business Model Documentation

A top-down approach would have gathered the type of data described in the 'AS-IS' business model from first from subject matter experts and secondly from existing system assets. This approach normalizes data from the outset, but delays the issue of mapping existing data to new data until the physical database is designed. For system migration efforts and DSS development efforts, this often delays the discovery of many errors and misunderstandings. With the reverse data engineering approach, the data are continuously mapped, error discovery is accelerated, and normalization is accomplished as the data definitions in the existing system are clarified. Each focus session with the subject matter experts involved a small subset of the data model that could be fully explored by the staff members involved in the discussion. As sections of the Business Model documentation were completed they were next analyzed for complete normalization and integration into a comprehensive model.

A.3 Develop 'TO-BE' Normalized Model

The data modeling team developed the 'TO BE' normalized model by applying normalization rules to each increment or subset of the Business Model completed. Figure 4 illustrates one example of the dramatic change in the model that took place as a result of analyzing the domains of values included in the Account Entity (i.e., the UFETS013 table in the physical model). The logical model as presented in this paper retains physical attribute names to support traceability. The UFETS013 table stored information on over

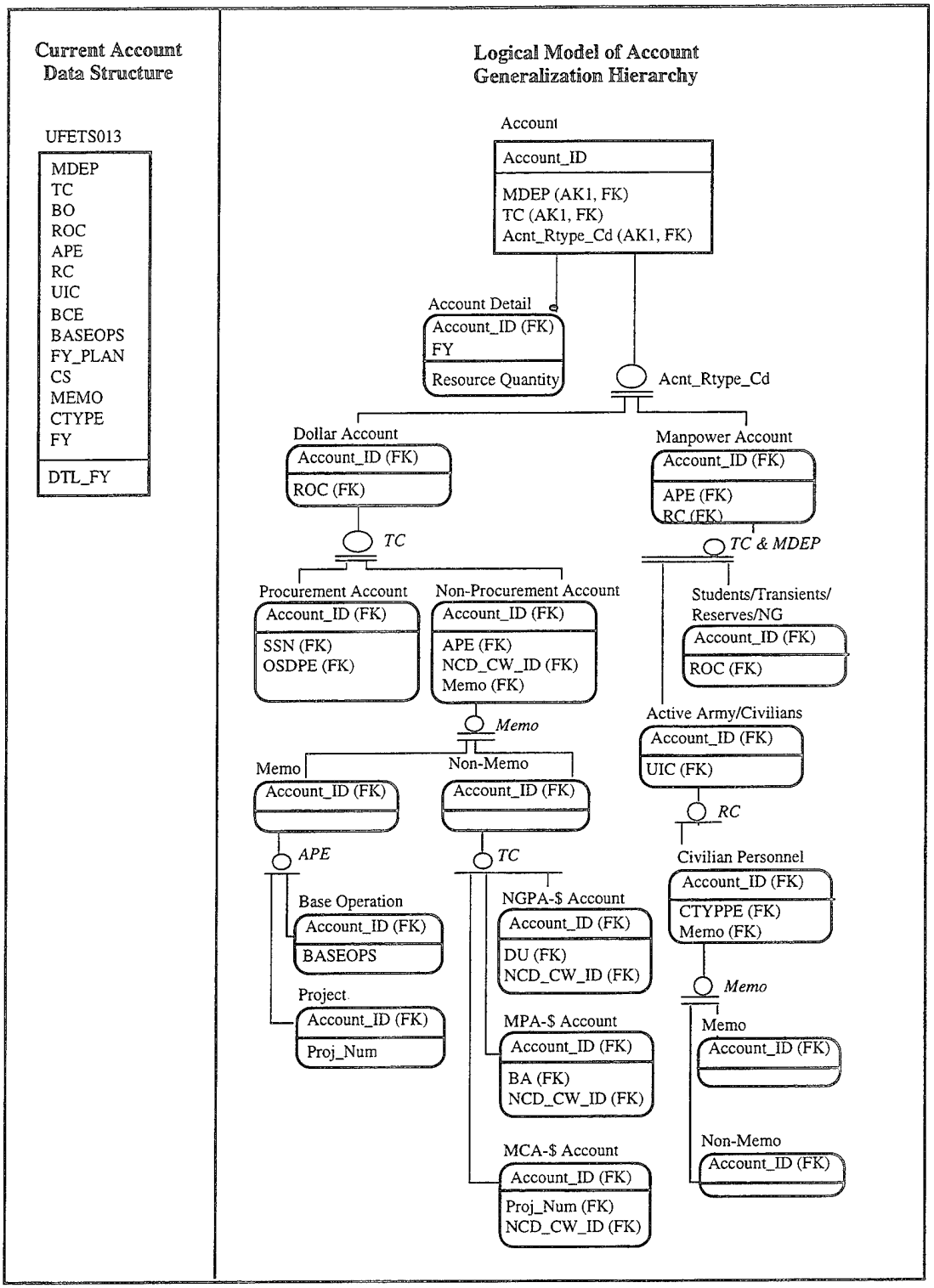


Figure 4: Current and Modeled Views of Probe Account Detail Data

forty different account types. The model on the right shows that there are at least twelve cases where moving from one account type to another causes a new or different set of attributes to be captured. Knowledge of what different attributes are needed for different categories of accounts is hidden when the model on the right is collapsed into a single table structure shown on the left. Furthermore, the key for uniquely identifying an instance of data changes for different types of accounts in the physical system, and these differences are also hidden when all data are collapsed into a single table.

Normalization of the model in light of the domain analysis completed introduced new perspectives that again caused the team to go back to the subject matter experts to further resolve issues. For example, the model on the right restricts itself to a single surrogate primary key called Account ID. This single key carries no intelligence for the type of account represented in any category, allows different types of accounts to be characterized by different combinations of elements, but represents a fundamental change in the way accounts are managed within the system. Could interfaces to other legacy systems be maintained? Could a reasonable screen dialog be designed? The answer to both questions was yes, and the Account ID was established as a unique number assigned to each account.

Note: An interesting IDEF1X data modeling syntax anomaly arose in representing Account generalization hierarchy in Figure 4. Treasury Code (TC) and Management Decision Package Code (MDEP) are used several times as inherited category discriminator attributes. IDEF1X only allows inheritance to be passed sequentially from parent to child without skipping levels as is practiced in this model. We encountered no negative impacts for breaking this rule in the Account generalization hierarchy shown in Figure 4.

A.4 Integrate Model

The normalized subsets of data were integrated into a comprehensive model. Because the scope of this effort involved a single data modeling team there were no major issues concerning resolving synonym, homonyms, or discrepancies in the level of abstraction used to represent the data. However, the integration brought issues concerning relationships between the subsets to the fore, and these were validated by tracing back through the data model to the original database structure, running queries designed to test the teams understanding of the relationships, and validating the results of these queries with subject matter experts. Additionally, findings of the domain analysis were reviewed in several data model walk-throughs conducted with selected representatives of the Program Analysis and Execution (PA&E) Directorate.

A.5 Develop the Physical Database Design

When performance implications were considered for the logical data model, the model was denormalized to efficiently support Probe transaction processing and system interface requirements. As shown in the side-by-side comparison of the logical and physical view in Figure 5 of the Account Generalization Hierarchy reviewed earlier, the twelve categories have been reduced to five categories. From the perspective of this paper, the specific rationale for the changes designed into the physical database design are not important. The major point is that from the perspective of system users, and from the perspective of the DBMS that manages the database, some of the rules the data modeling team took time to uncover are now hidden once again.

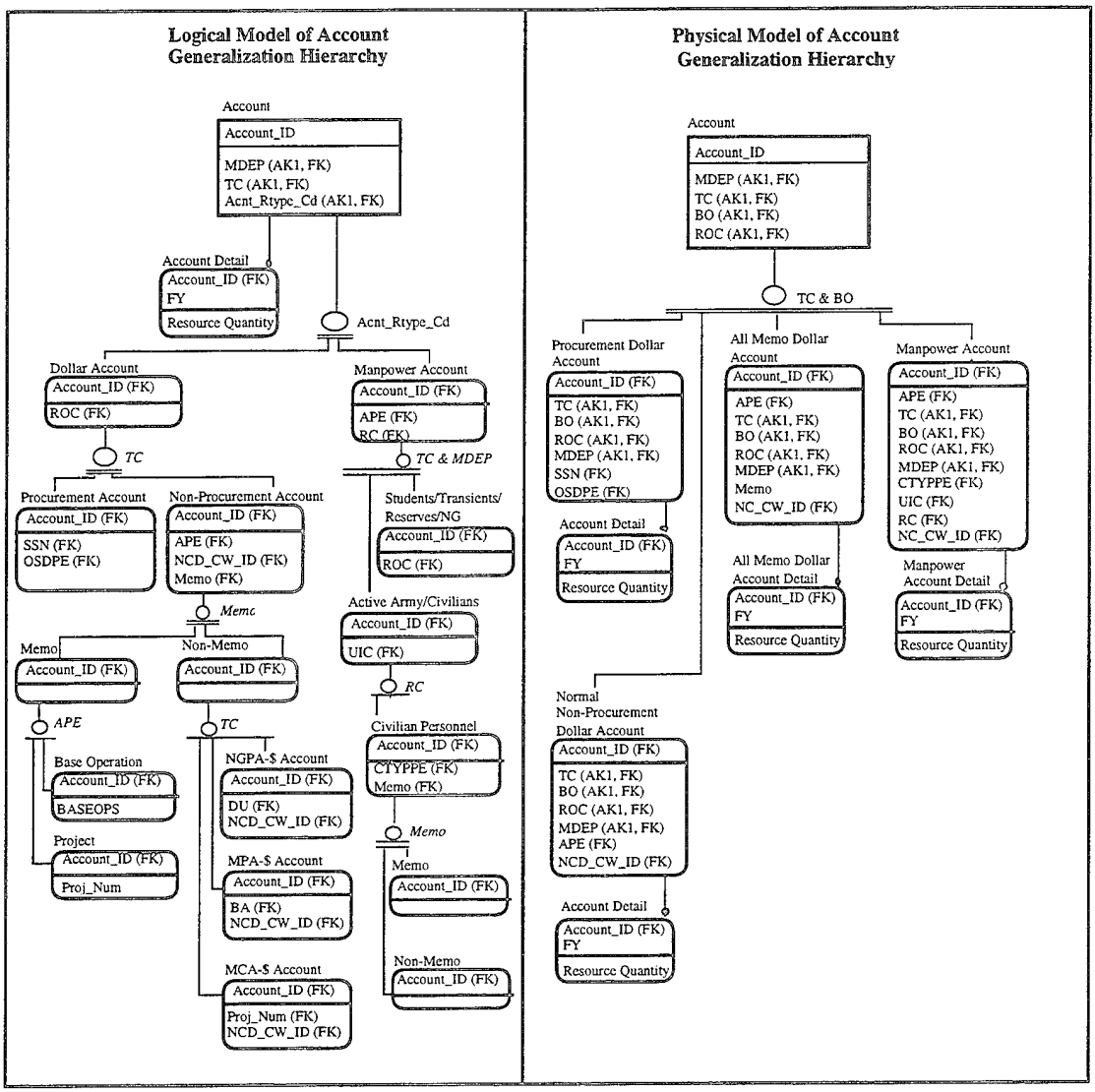


Figure 5: Logical and Physical Views of Probe Account Generalization Hierarchy

All to often these rules stay with the CASE tool, eventually are archived, and gather dust as an archived testimony to the wonderful analysis conducted by a hard working team of data modelers, subject matter experts, and system developers. Under better practices these rules are exported to a centralized repository for a corporate data model. But 'best practices' should seek to put these metadata to work in the actual system so that the business rules remain visible to a wider audience than those capable of mapping logical models to physical models or those capable of reading software code. The following sections explore in more detail the potential benefits of metadata captured while performing domain analysis.

B. Potential Benefits of Domain Analysis Metadata After Design and Development

Domain analysis metadata need to be captured for any data modeling efforts requiring reverse engineering of exiting data. Examples of this type of situation include conversion of data from legacy systems to new systems, or extraction of data from

existing systems for employment in new uses with DSS applications. The history of changes made to most legacy applications alone is bound to have caused many data requirements to be implemented in ways that make the business rules invisible to the system user. Experiences of the data reverse engineering effort conducted for the Probe are the norm for those that practice data modeling and domain analysis as synergistic disciplines. Currently there are two problems associated with capturing this level of metadata:

- CASE tools seldom collect the mapping rules for converting data from the legacy system to the target system. These metadata maps are very often maintained in spreadsheets that do not get cataloged for reuse.

- Because there is no appropriate way to store this information in the CASE tool, business rules captured as relationships among data elements or domain values are not represented in any centralized corporate repository. Rather, they are represented in the 'IF-THEN-ELSE' logic of application programs.

Data standardization is one practice that normally encourages the marriage of data modeling and domain analysis disciplines. In most general terms, data standardization programs attempt to improve system interoperability and data sharing by establishing a unified definition of the data -- which should include the domain of allowable values. Unfortunately, data standardization efforts often treat every data element equally and leave little or no room for discretionary risk management. They seldom, for example, discriminate between the rigor applied to data elements that are used as primary keys versus elements used only to carry textual descriptions. Furthermore, they often put more emphasis on naming conventions, definitions, and data stewardship identification, than on domains of allowable values. Figure 6 lists metadata requirements for the DoD Data Element Standardization program, identifies which of these metadata items were used in the data reverse engineering project for Probe, and briefly describes why those used were important.

It's important to recognize that the Probe data reverse engineering had parochial interests for managing metadata. Those elements listed as requirements for the DoD Data Standardization Program in Figure 6 that were not used in the Probe Rightsizing project for the most part are required to support the global goals for review, approval, and reuse of the standard data element specifications across DoD. If the metadata are exported to a centralized corporate repository such as the DoD Data Repository System (DDRS), other system designers can reuse this information, but what about the local interests of the Probe system?

There are benefits beyond the improved quality of the design and development effort. These benefits relate to making the business rules applicable to Probe data visible to the end users and easy for the application maintenance/data base administration staff to maintain. The normal fate of business rule information at the domain level is that they get imbedded in IF-THEN-ELSE logic of application programs. The only visibility of the business rule to end users are the error messages they get while performing on-line maintenance functions. Furthermore, the rules get duplicated throughout the system for any application using the business rule. As a result, application maintenance staff are afraid to touch the code.

Knowledgebases provide facilities for expressing business rules and using them for a variety of applications. Unfortunately, most transaction systems use database management systems (DBMSs) rather than knowledgebases. If a DBMS provides

Metadata Required for DoD Data Standardization	Used by Probe Project	Usage Description for the Probe Rightsizing Project
Prime Word (Entity)		
Name	°	Entities were reverse engineered and had to be assigned business names and definitions to improve understanding among modelers and users.
Definition Text	°	
Functional Area Identifier		
Proponent Model Name		
Standard Data Element		
Name	°	Data Elements had to be assigned business names and definitions to improve understanding among modelers and users.
Definition Text	°	
Data Type Name	°	
Maximum Character Count Quantity	°	Characteristics of type, length, decimal count, unit of measure, and formula definition needed to be captured to migrate the data to the new system environment.
Decimal Place Count Quantity	°	
Unit Measure Name	°	
Formula Definition Text	°	
Prime Word Name		
Functional Area Identifier		
Derivation Type Name		
Authority Reference Text		
Access Name	°	New access names were assigned to elements migrated tot he new system environment
Automated Information System Name	°	Linkages to the old system as well as interfaces to the old system had to be maintained
Automated Information System Identifier	°	
Non-Standard Data Element Name	°	
Functional Access Name		
Domain		
Low Range Identifier	°	Domain information was critical for uncovering hidden business rules and completely understanding the meaning of the data element.
High Range Identifier	°	
Source List Text	°	
Domain Value Identifier	°	
Domain Value Definition Text	°	

Figure 6: Metadata Requirements for the DoD Data Standardization Program Compared to Metadata Required to Support the Probe Rightsizing Project

facilities for defining referential integrity rules at a field level or a table level, this helps reduce duplication of the business rules and ensures that all create and update transactions abide by these referential integrity rules. The problem here is that the rules are only applied to maintenance functions, and are only visible through the error messages displayed when the end user offends the rules specified during create, update, or delete operations. They cannot, for example, be used to help the user find data they need.

A metadata mapping from the logical to the physical database design would make business rules visible and table driven so that they are easy to maintain. However, commercial tools providing this type of mapping are generally expensive products oriented towards accessing data across heterogeneous platforms (e.g., InterViso by Data Integration, Inc.). Building this capability in-house is even more expensive. Furthermore, many of the business rules extend below the entity relationship mappings and data element mappings well supported by these tools. Domain associations are not well supported by existing commercial tools designed to support generalized data access across heterogeneous platforms.

One other alternative that is explored and documented in this paper is the option of documenting business rules that are represented as associations among data elements and their domain values in relational tables. This metadata can then be used for a variety of purposes, to include:

- Govern data edits based upon rules appropriate for different business rules (e.g., accounts type specification)
- Determine what type of account a specific record represents
- Help a user navigate through the database to find a record of a given account type.

The data model shown in Figure 7 captures business rules in entities that are divided into two groups: 1) Business Rule Type and Component Description, and 2) Business Rule Condition Specification.

B.1 Business Rule Type and Component Description

Six entities in the upper portion of Figure 7 describe two categories of metadata: 1) the different types of data element associations that are managed under Probe -- to include business rules, as well as the attributes required for each type of account, and 2) the constituent object types and relational operators for defining business rule conditions applicable to the various types of accounts.

The business rules are described in the *Association* entity simply as a type of data element association using functional terms (e.g., Procurement Dollars Account Rule, Civilian Personnel Account Rule, Active Army Unit Account Rule). The attributes required to characterize each account type are listed in the *Association Member* entity, and must be defined in the *Attribute* entity.

Edit conditions for validating different Account Types are assembled from objects (e.g., subdomains of attribute values) described in the *Object Type* entity, and relational operators (e.g., '=' or '>', or 'in') described in the *Relational Operator* entity. Allowable combinations of object types and relational operators are listed in the *Relationship Object Type* entity. For example, the relational operator 'in' is appropriate for specifying that an attribute value is in a 'subdomain' object type, and the relational operator '=' is appropriate for specifying that an attribute value equals a 'constant' object type.

B.1.2 Business Rule Condition Specification

Six entities in the lower portion of Figure 7 specify sets of business rule conditions for validating different types of Probe data (e.g., Probe accounts). A condition can be viewed as IF conditions (e.g., IF Budget Obligation (BO) = 1 THEN...).

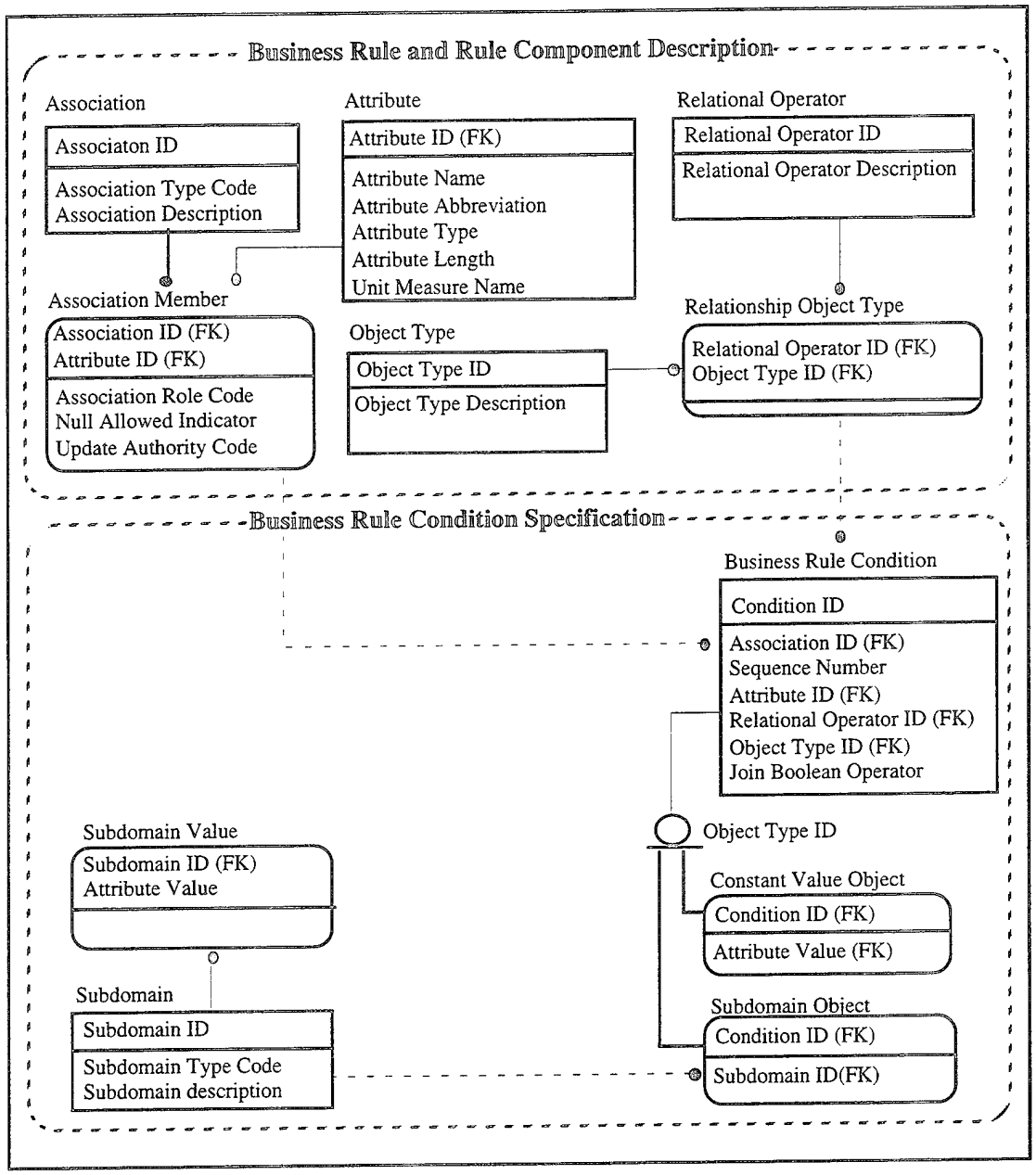


Figure 7: Business Rule Specification Metadata View

The conditions can be joined into groups of conditions in the *Condition Join* entity. These joins of rules are analogous to compound conditions in an SQL WHERE clause joined using AND/OR boolean operators. The rules can be based on relational operations against several types of objects. Two types of objects are explicitly depicted in the model: 1) *Constant Value Object,* and 2) *Subdomain Object.* A constant represents

a single literal value included in the rule. A subdomain, in contrast, can be defined as a list or range of values. Other types of rule objects, such as software edit procedures, can also be defined to support validations of the various types of Probe Accounts.

B.2 Entity Definitions for Business Rule and Component Descriptions

Association- identifies and describes the different types of data element associations encountered within Probe, to include Probe account type business rules (e.g., Procurement Dollars Business Rule might have the label 'Proc-$' assigned as an identifier). One record exists for each business rule defined as well as any other type of data element association.

Attribute - names and describes the attributes characterizing different types of accounts. These descriptions include general domain characteristics such as attribute type, length, and unit of measure. Other attributes can be added if appropriate (e.g., picture format, base domain reference). One record exists for each attribute created and/or used by Probe system.

Association Member- correlates attributes to specific business rule associations, specifies general edits applicable for the element's participation in a business rule (e.g., specify whether or not a null value is allowed for an element participating in a business rule association). One record exists for each valid combination of Attribute and Association.

Object Type - identifies different types of objects (e.g., "Constant", "Subdomain", "Procedure") that can be used in condition statements to specify validations for attributes used in different types of accounts.

Relational Operator - identifies and describes the different relational operators (e.g., "=", ">", "<", "<>", "<=", ">=", "in", "not in") appropriate for using in conditions with one or more object types listed in the object type entity.

Relationship Object Types - correlates relational operators with the appropriate object types for building account validation conditions. One record exists for each valid combination of an object type and relational operator.

B.3 Entity Definitions for Business Rule Condition Specifications

Business Rule Condition - Specifies one or more validation rules for an attribute's participation in a specific business rule. One record exists for each application of a condition to a specific business rule. Each condition specifies a relationship between an attribute and a validation object. For example, the statement "BO = 1" decomposes into the following constituent parts:

- Attribute ID "BO"
- Relational Operator "="
- Object Type the constant "1"

The Object Type ID provides an extensible code for referencing a wide variety of object types such as subdomains, special edit procedures, and other attributes. The model currently provides explicit representation of 'subdomain' and 'constant object' types. Others could be added as they are required. If several conditions need to be joined, the attribute named Join Boolean Operator specifies how using boolean operators (e.g., "AND", "OR", "XOR").

Constant Value Object - identifies the value of a constant referenced as the object of a validation condition.

Subdomain Object - identifies a specific subdomain of values referenced as the object of a validation condition.

Subdomain - describes a list of valid values for an attribute within the context of one or multiple business rules. A single subdomain can be referenced by multiple conditions for different business rules, and a set of conditions for a single account type attribute can reference multiple subdomains (i.e., subdomains can be linked for use in validating a single account type). A subdomain can be defined as being one of several types (e.g., a list of included values, and, minimum and maximum specified for a range of included values).

Subdomain Value - lists values for each subdomain. The values may represent an enumerated list, or minimum and maximum values for a range of allowed values.

B.4 Example Business Rule Specification for an Account Type

A simple example of a rule for identifying procurement accounts will illustrate how the data model depicted in Figure 7 is populated and used. Figure 8 presents example values for specifying the following rule for Procurement Accounts:

BO = 1 AND TC in ('2031', '2032', '2033', '2034', '2035')

To conserve space in the exhibit, some attributes are omitted from the example, but values are depicted for all fields needed to illustrate how the entities are related. The procurement account type business rule is identified as "PROC$" in the *Association* entity, and the BO and TC are identified as attributes used to describe a procurement account through the *Association Member* entity. The rule specified above is divided into the following two conditions in the *Condition* entity:

BO = constant	(Condition 00111)
TC in Subdomain	(Condition 00112)

The specific constant for the first condition (i.e., '1') is recorded in the *Constant Value Object* entity, and the Subdomain (i.e., 'PROC TC') is recorded in the *Subdomain Object* entity. The values for this domain are found by following the relationship linking the *Subdomain Object* entity to the *Subdomain* entity (which indicates that PROC TC is an enumeration of values rather than a range), and then to the *Subdomain Value* entity. The *Subdomain Value* entity lists the set of values identified in the rule specified above.

The two conditions for the rule are joined in the *Business Rule Condition* entity. Because the 'AND' join boolean operator is specified, both conditions must be met in

order for the account to represent a procurement account. If the 'OR' operator had been used, only one of the conditions would have needed to be satisfied.

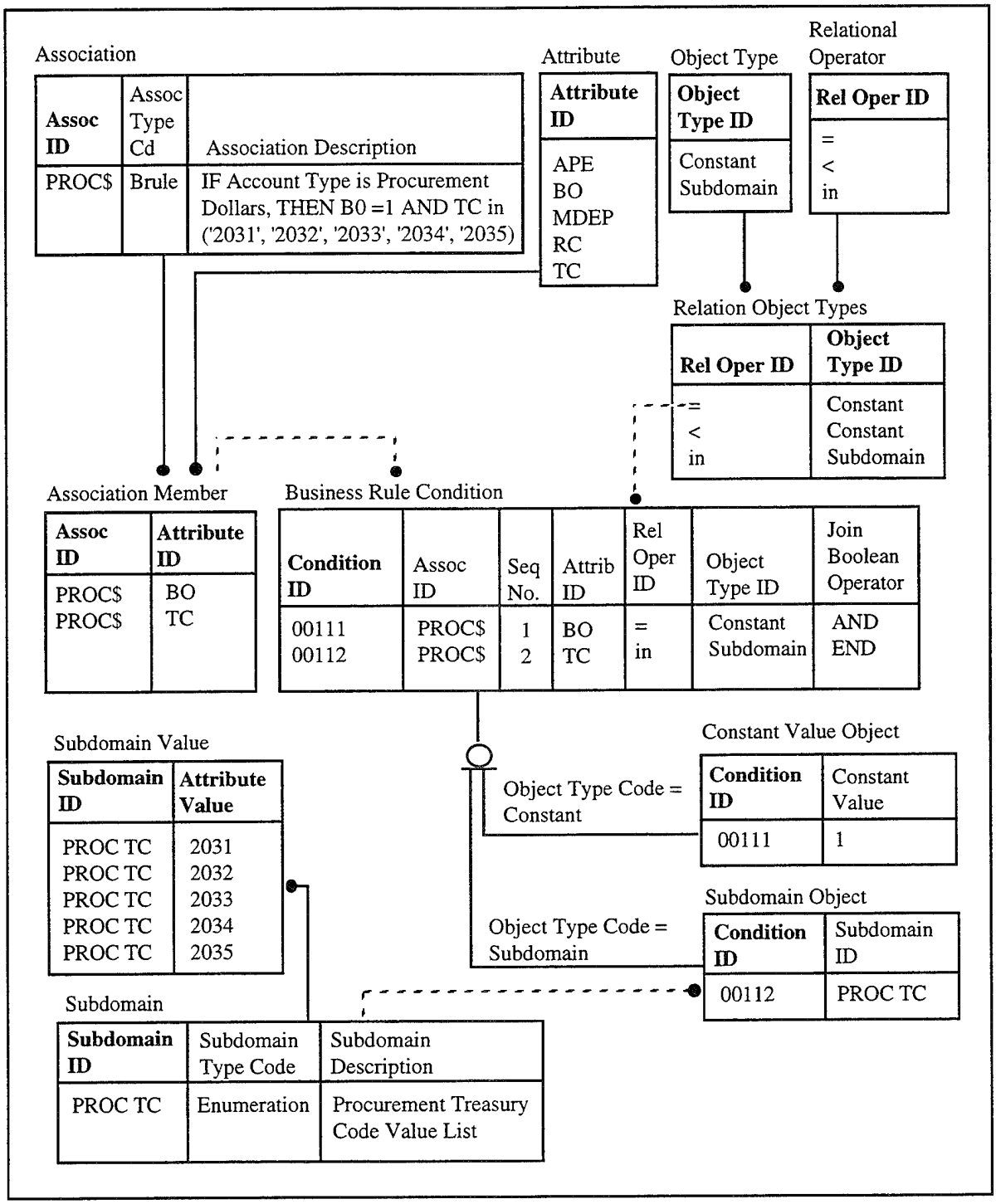


Figure 8: Example Account Type Specification Business Rule

C. Benefits of Domain Analysis Business Rules for DSS Architectures

The previous sections described the requirements for capturing metadata describing associations between data elements and domain values to support system migration projects, and the benefits for using this metadata to support day-to-day system operations. This section describes how similar requirements and benefits exist for DSS architectures.

C.1 Description of the DSS Architecture and Metadata requirements

Figure 9 graphically depicts the overall structure of data, communications, processing, and presentation capabilities required for supporting end user DSS applications using a data warehouse architecture. Metadata captured from CASE tools, database catalog schemas, as well as file layout descriptions in application programs are essential for supporting the objective for universal data access in a DSS architecture. The architectural layers for the DSS architecture shown in Figure 9 and their metadata requirements are briefly described below:

- Data Supplier Layer: The data supplier role for a data warehouse is typically supported by transaction systems designed to support day-to-day operational processing. These systems were developed to efficiently process a well defined set of transactions (e.g., add a new employee to the personnel database, document a purchase, and assign an employee to a project). Because of the limited focus of the operational processing supported, the DSS uses for the data were not anticipated or planned. As a result, these databases are difficult to access for DSS purposes. Furthermore, to access data across a number of data supplier systems the user must interrogate data on a number of different hardware/software platforms that maintain their data in disparate databases (i.e., related data across these systems cannot be easily joined).

- Data Access Layer: The data access layer comprises middleware that allows DSS Application Tools to access data from the data warehouse, and allows the data warehouse to access data from the data suppliers. Middleware products such as EDA/SQL, CrossAccess, and InterViso make it possible to use SQL to access a multitude of relational and nonrelational databases/ data file management systems within the data supplier level. These data access middleware systems span not only a wide variety of DBMS products, but also across a wide variety of manufacturer protocols and network protocols. Metadata for these tools typically represents network addressing information, physical database schema descriptions, network protocol information, and limited data transformation rules.

- Import Processing Layer: The Import Processing Layer represents applications that regularly copy or replicate data from supplier databases to temporary storage areas, where the data are then prepared for importing into the data warehouse. These applications select, validate, merge, aggregate, and load the data warehouse with information accessed from

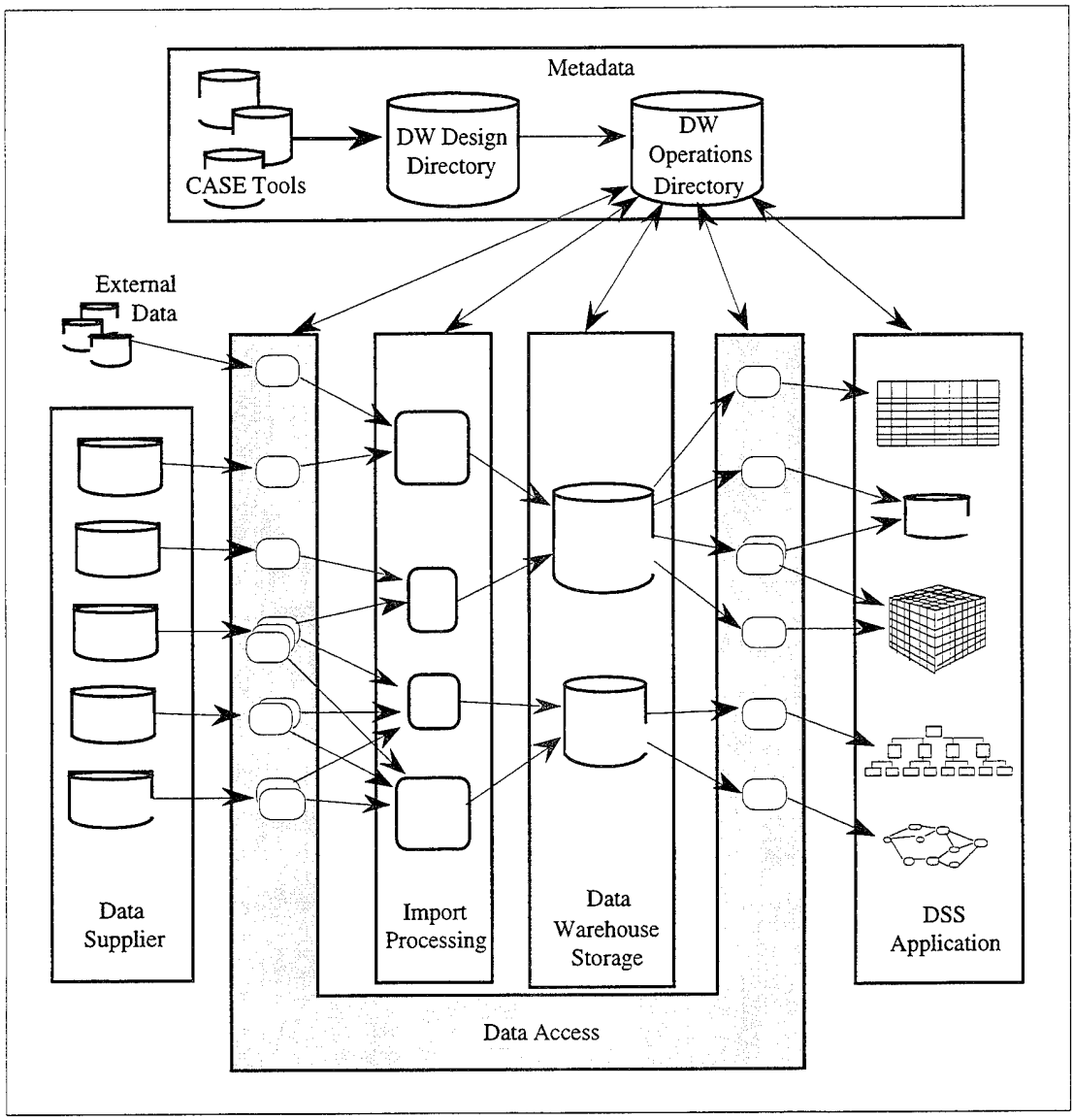


Figure 9: Data Warhouse Infrastructure Template

the supplier systems. They also archive obsolete data and develop standard historical extracts for customers. Metadata for these tools include a significant amount of metadata describing:

-- Schema structures for databases and images of extracts being imported into or extracted out of the data warehouse

-- Mappings of data copied from source data stores to target data stores, to include any temporary files used

- -- Transformations of data that go through format changes during the extract processing

- -- Validations used to measure the quality of the data imported into the data warehouse

Example extract management and data transformation tools for import processing include PRISM Solution's PRISM Data Warehouse Manager, Carleton's Carleton Passport, and Evolutionary Technology's Extract Tool Suite. Example data validation tools include QDB Solution's QDB analyze, and Vality Technology's Integrity Data Re-engineering Tool.

- Data Warehouse Storage Layer: In the Data Warehouse Storage Layer, data selected, integrated and certified for supporting DSS applications are stored in a form that is flexible and easy to access. In cases where there are several levels of data warehouses, subject area warehouses will typically format the data with more emphasis on integration and flexibility for accommodating future access, while departmental databases will format the data with more emphasis on data legibility and easy access. There are database management systems specialized for addressing data warehouse issues related to achieving good read and load performance with large databases (i.e., billions of records involving storage in the 100 gigabyte to terabyte volume range). Examples include Red Brick's Data Warehouse Management System. Database management systems built upon parallel processing architectures also promise high performance for both read and right access to large volumes of data. Examples include Tandom's Himalaya series of parallel processors, IBM's SP2, Pyramid Technology's Reliant RM1000, the Sybase Navigation Server, ORACLE (version 7.0) and Informix (version 7.0) with the Informix On-line Dynamic Server. The DBMS used to store the data warehouse typically includes a catalog describing the physical schema for the data warehouse.

- DSS Application Layer: The DSS Application Layer represents specialized tools for end users to acquire, transform, analyze, and develop presentation materials in support of completing decision making tasks. Example tools include query formulation tools such as Business Objects, and Clear Access; spreadsheets such as Excel; multi-dimensional databases such as IRI's Express; artificial intelligence application development tools such as AICorps KBMS and AION's ADS. In recent years the portfolio of DSS development tools to choose from in the commercial market has grown, and can be expected to continue to grow in the future. Many of the new tools require metadata to operate effectively, and tap into the metadata provided by commercial DBMS catalogs housing the data warehouse. These metadata are designed to help the user find the data they require from the data warehouse and transform the data into the proprietary database used by the tool.

Currently there is no single tool capable of supporting the DSS architecture described above. Current practices for designing and developing DSS architectures primarily involve lashing several tools together. The most troublesome aspect of this approach is the requirement to redundantly enter metadata required for each of the separate tools. Currently there are no open standards adopted among the tools commercially available for sharing metadata.

C.2 Metadata Requirement Shortfalls for the DSS Architecture

Managing data and supporting customers for a DSS architecture requires a variety of metadata: data about the customer views of data, the suppliers of that data, processes used to extract, validate, load and archive the data, and rules for synchronizing the data. It's metadata that will allow end users to access data from the data warehouse without having to know where the data are stored, the format in which the data are stored, or the tool that delivers the data to the screen. Metadata documenting the data stored at any level of the DSS architecture must be mapped to metadata for the same data at any other level of the DSS architecture. This mapping allows the same data elements to be viewed in many different logical views depending on their usage.

A DSS architecture based on a data warehousing can be viewed as a continuous steady state of data conversions to support new DSS applications. The data conversion is designed and performed once in a system migration effort, but is performed over and over on a regular periodic basis in a DSS architecture. This suggests that if a large practice is being made of system migration as is currently the case for DoD, then many of the tools used to support data warehouse architectures may provide a significant return on investment for the system migration effort. In fact, some tools marketed for employment with data warehouses actually have their origins as solutions for supporting data conversion efforts in large IS shops with lot's of system enhancement and database conversion work (e.g., Evolutionary Technology's Extract Tool Suit).

Metadata used by commercial tools currently available for supporting data warehouse operations largely adhere to the entity-attribute-relationship paradigm for representing data structures. This gives rise to the most prevalent problem facing users; these tools fail to address the more difficult and complex data transformations and edits designers and developers face when confronting data extracts from legacy systems. In short, the existing tools do not capture metadata concerning business rules described through associations among data elements and domain values and so cannot put this metadata to work in DSS architecture operations. Very often the more complex business rules once again are implemented using IF-THEN-ELSE logic within subroutines called from exits provided by the commercial software products.

D. Summary and Concluding Remarks

Traditionally the disciplines for software reuse and data asset reuse have been managed as disciplines to promote systems interoperability and reduce systems development and maintenance costs. Example data assets commonly managed for reuse include data models and data element standards. Example software assets commonly managed for reuse include software modules and generic software architectures for recurring software system requirements (e.g., on-line table maintenance, extract generation).

Business rules represent a potentially valuable reusable asset that are currently captured as documentation accompanying data models, but then implemented using "IF-THEN-ELSE" logic in software. Furthermore, the business rules currently captured in data models based on relationships between entities represent a subset of the business rules required to manage most applications. There are more complex business rules that also need to be coordinated that cannot be represented using standard entity relationship diagramming techniques; these can only be represented by describing the associations

that exist among data elements and data element domains. Systems analysts and programmers are often reticent to change a large system because they do not know the total impact of the change when business rules are intertwined throughout the software code as "IF-THEN-ELSE' logic. Business rule reuse practices are likely to streamline the analysis and coding work required for software maintenance activities.

A business rule reuse practice may represent the cornerstone for bringing together disciplines for software and data asset reuse. Software and data asset reuse disciplines can be merged during system software development by developing metadata governed software. If this practice is adopted as a standard, the benefits of capturing metadata to support data asset reuse will extend more rapidly into the realm of the day-to-day system operations. Business rules for managing data will become more visible to the users, and corporate awareness of business policies and practices will improve.

Report generation tools using SQL provide a useful analog for understanding the potential merits of business rule reuse for managing data and software assets. SQL has provided a framework to standardize interfaces for retrieving information from databases. This, in turn, has made it possible to develop open architectures for systems to allow plug-and-play compatibility among commercial report generation products and relational DBMS products. In a similar fashion, standardizing the syntax and metadata used to represent and communicate complex business rules could provide a framework for developing generic middleware software products. These products could also be plug-and-play compatible across a variety of relational DBMS products, and reduce the volume of software that needs to be developed, tested, and maintained to support projects to migrate application system as well as develop DSS architectures.

Data administrators need to proactively research methods for extending entity relationship modeling (e.g., IDEF1X) syntax to capture complex business rules, and explore the potential value for reuse of business rules to support common types of software development requirements. System migration efforts that require business rules be applied across the legacy systems being consolidated represent excellent pilot cases for two types of generic applications:

- **Data Quality Assurance** -- applications to extract samples or entire copies of data from one or multiple databases and run data quality assurance tests against the data using business rules cataloged in a business rule repository.

- **Business Rule Governed Software** -- software either generated and compiled with business rules embedded as read from a business rule repository, or designed to interpret the business rule repository at execution time. A specific types of applications include interactive data entry and extract generation.

The objective is to develop middleware for enabling business rule reuse by fitting the software and data assets to a common reusable architecture encapsulating software and business rule metadata. As a modernization initiative, this type of middleware offers potential for significants by:

- Improving the level of modularity and reuse achieved in system migration efforts

- Incrementally reducing the volume of software that needs to be specified, developed and tested for each new application system

- Accelerating the achievement of system interoperability
- Improving consistency of information provided to decision makers
- Reducing the level of work required to manage software configurations in response to business rule changes
- Reducing the amount of time knowledge workers spend resolving discrepancies in information provided from multiple sources

Enhanced modeling notation along with procedures and tools advancing the management of business rules for reuse will improve software development and maintenance practices. Ultimately business rules will migrate from embedded elements of each unique software application to standard inputs for middleware used in common.

Duane Hufford is a Senior Principal technical consultant for American Management Systems defense clients. He has worked on a variety of systems development and data administration projects for DoD, Army, and Navy organizations focusing on improving the documentation about data, making data more accessible, and improving data quality. These include leading design and development of dictionaries to support data standardization and to support management of data warehouses, conducting training on data administration concepts and procedures, and helping organizations develop data administration policies and procedures. He commonly speaks at professional gatherings and symposiums concerned with data administration and has written articles for publications such as Auarbach and Database Advisor. Duane can be reached at (703) 841-6066.

Managing Change Within Federated Database Schemas

Christopher J. Bosch

The MITRE Corporation

1. INTRODUCTION

Over the past decade we have begun to witness a fundamental shift in the way in which information systems are developed. That shift is leading us away from developing separate information systems which address specific problems and toward integrating existing information systems with new applications in enterprise-wide solutions to our problems. Our goal is for information systems within an enterprise to be *interoperable* as required. Each system should be able to get the information it needs from other systems, and each system must provide information in a form that can be interpreted and used by other systems.

The idea of a *federated database system* has been advanced as a means of integrating and promoting interoperability among distributed, heterogeneous databases [Sheth and Larson, 1990]. One approach to establishing such a federated database system involves *encapsulating* local databases using a *common object model* as shown in Figure 1. In this figure, each gray oval represents a set of classes defined using the common object model. Each *component schema* represents the object-oriented encapsulation of a component database's *local schema* which is expressed in some other data model (e.g. relational, hierarchical, network). One or more *export schemas* can be defined for each component schema, and then *federated schemas* are composed from the export schemas of two or more other databases (which may be either local or federated).

The degree of *autonomy* which components of integrated information systems can exercise is always an issue. Simply put, certain types of actions initiated autonomously serve to magnify the problems due to distribution and heterogeneity. One area in which some autonomy might be exercised by the components of a federated database system is in the design and evolution of their component and export schemas. In a federated database system, it will not be the case that some strong central authority will dictate the design of components' schemas in a top-down fashion. Rather, schema design and evolution will proceed as a result of actions initiated both from the "top" (some central authority at the federation level) and from the "bottom" (organizations or individuals responsible for the component databases).

At MITRE, we are investigating tools and techniques that will allow organizations participating in a federation to manage change within the federated database's component, export, and federated schemas. In this paper we report the status of our efforts to date and indicate future directions for our work. Section 2 describes the problem of schema evolution for a hypothetical federated database system, and section 3 details the technical approach we are employing to solve that problem. Section 4 then gives an overview of the environment for managing change we are developing to address the problem of schema evolution in federated database systems. Finally, section 5 summarizes the material presented in this paper and indicates future directions for our work in this area.

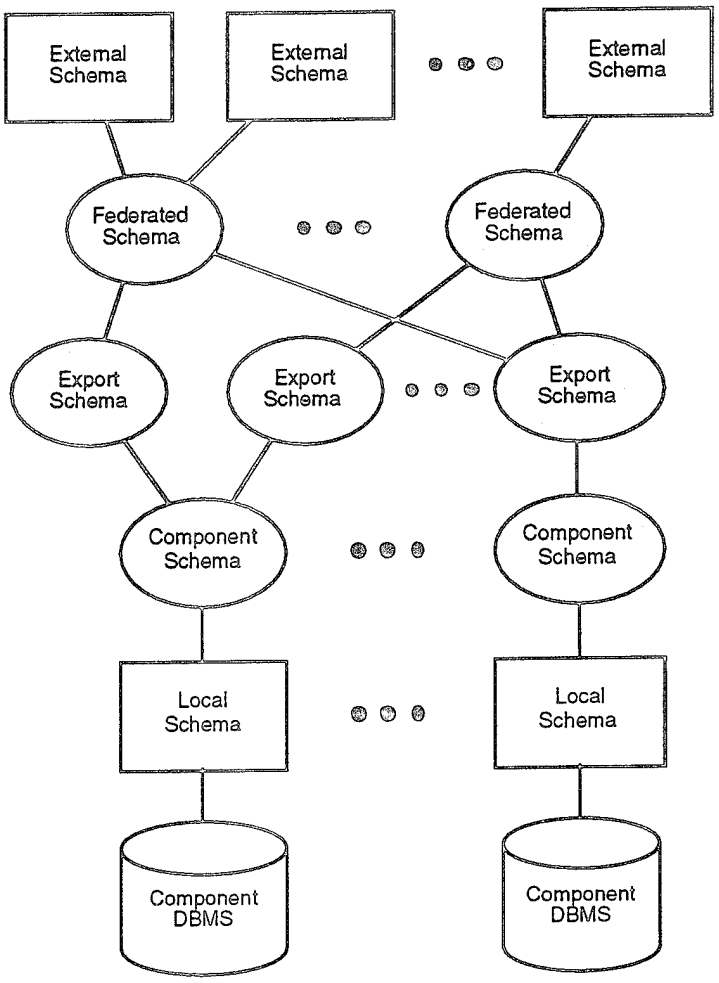


Figure 1: Federated Database Architecture

2. PROBLEM DESCRIPTION

To illustrate the problem of schema evolution in federated database systems, we will describe a hypothetical federated database system that could be developed to provide for data interoperability among various C2 databases within the Global Command and Control System (GCCS) framework. Currently, there is a great deal of emphasis being placed on promoting interoperability among C2 applications within GCCS. The GCCS Common Operating Environment (COE) provides a standard set of application programming interfaces (APIs) to be used for developing C2 applications, and use of these standard APIs will provide some degree of application interoperability. Because C2 applications will continue to interface with databases defined outside of the COE as shown in figure 2, the problem of data heterogeneity will remain as an impediment to sharing data between these C2 applications.

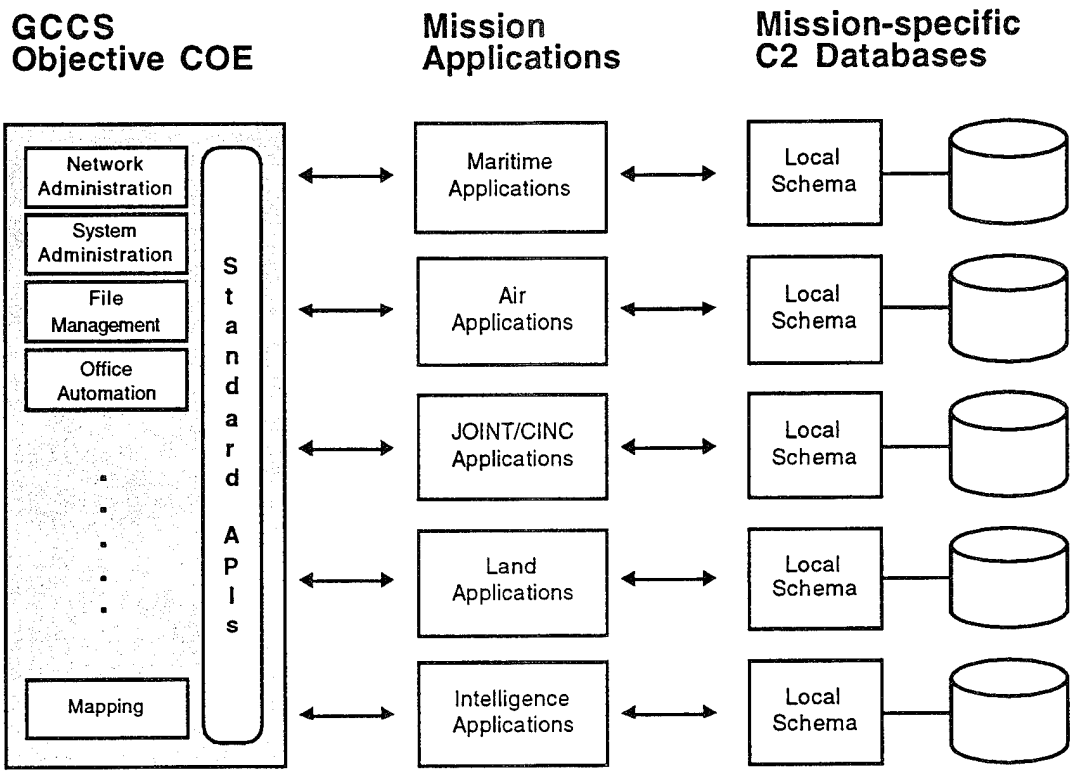


Figure 2: GCCS Applications and Databases

An approach to providing for data interoperability among C2 applications that is being considered within the GCCS Leading Edge Services (LES) program involves development of a federated database which integrates the heterogeneous data types managed by various C2 databases. This approach, shown in figure 3, will in effect provide a set of APIs for accessing and manipulating data managed by the C2 databases. Unlike the standard APIs provided by the GCCS COE, however, these interfaces can be expected to change with greater frequency.

Schema changes in this hypothetical federated database system will be required due to a number of reasons. As different mission-specific C2 databases are brought into the GCCS framework, component schemas encapsulating those databases must be developed, and the federated schema will require changes to accommodate the new component databases. Introduction of a new C2 database into the GCCS framework may even call for changes to be made to the component schemas of C2 databases previously encapsulated. For example, if we were to discover conceptually similar data in those other databases and wanted to create a generalized class in the federated schema expressing that concept, we would then have to modify classes in the component schemas so that they inherited from the newly-defined parent class. Evolving mission application requirements may also dictate changes in the federated schema that could in turn require changes to the component schemas.

Whatever the impetus for change, one thing is certain. This set of interfaces — provided by the federated database system for accessing and manipulating C2 data — will change with greater frequency than the set of standard APIs provided by the COE. Whether changes to these interfaces are introduced in a top-down or bottom-up fashion, they all must be analyzed, designed, implemented, and tested without interrupting current services provided by the federated database system.

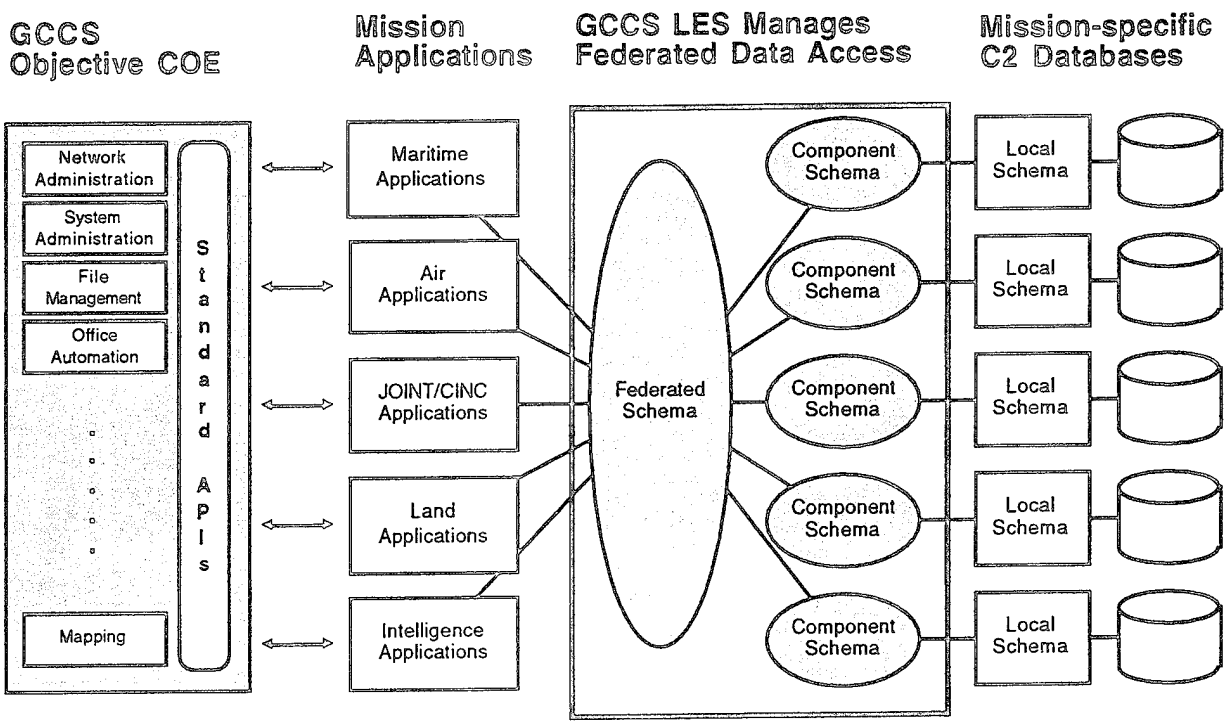


Figure 3: Federated Data Access in the GCCS Framework

3. TECHNICAL APPROACH

To address the problem of schema evolution in such a federated database system, we are developing a technical approach that integrates and extends established techniques from three distinct areas: *object-oriented schema evolution*, *version modeling*, and *dependency maintenance*. We summarize these techniques below.

Object-oriented schema evolution

In the past decade, much research has been done addressing the subject of schema evolution in object-oriented databases. Perhaps the most influential of the early works published on this subject have been those associated with the ORION object-oriented database management system [Banerjee et al 1987a, 1987b; Kim et al 1989]. The approach to object-oriented schema evolution developed by ORION researchers specifies the following items for the ORION object model:

- the invariants of schema evolution
- a taxonomy of schema changes
- a set of rules of schema evolution

By identifying a set of schema invariants, a taxonomy of primitive changes that can be made to a schema, and a set of rules which maintain those invariants as changes are made, the methodology put forth by ORION researchers guarantees that the new schema produced by applying primitive changes to a consistent schema will itself be consistent. This approach does not, however, provide

for alternative versions of either the schema itself or the classes from which it is composed. As such, it will not support the analysis, design, implementation, and testing of changes while continuing to support previously-defined interfaces. ORION researchers did later propose a model for versioning entire schemas [Kim and Chou 1988]; however, that proposal assumes a single database schema and it cannot be applied to a federated database system with its many component, export, and federated schemas being developed by separate, cooperating organizations.

The most important concept to be taken from research such as that done on ORION is the role that schema invariants play during schema evolution. Recently, some researchers have found the idea that schema invariants must be satisfied after each primitive change to be limiting in practice, and they have proposed complex schema-modifying operations made up of a number of primitive changes [Beldjilali 1995; Pons and Keller 1995]. In these proposals schema invariants still play the prominent role, only now they need not be satisfied until the end of each complex schema-modifying operation rather than after each primitive change is applied.

As can be seen by this recent work, object-oriented schema evolution is still an active area of research. Participants in a workshop on *Supporting the Evolution of Class Definitions* held in conjunction with OOPSLA '93 drew the following conclusions about the state-of-the-art with respect to class evolution.

> *It was agreed that our comprehension of the problem is still in its infancy. There were many similarities in the approach[es], and that no single magic bullet was going to emerge as a solution. Many of the approaches would seem to work well in combination. Clearly, having separation between interface and implementations provide[s] some help with the problem, but techniques and formal approaches still need to be sought.* [Goldstein 1993, p. 105]

Version modeling

Over the past decade many researchers have proposed approaches to modeling versions of objects in design databases. Katz reviewed the approaches proposed by these researchers and attempts to unify them by providing "a common terminology and collection of mechanisms that underlie any approach for representing engineering design information in a database" [Katz 1990, p. 375]. In his work unifying the various approaches to version modeling, Katz identified seven basic classes of mechanisms used in these approaches — these classes of mechanisms perform the following functions:

1. organize the space of versions
2. dynamically resolve references to specific versions
3. hierarchically compose a set of versioned objects
4. provide alternative groupings of versions
5. distinguish between instances and definitions
6. handle change notification and propagation
7. organize the set of workspaces

The environment for managing change within federated database schemas that we describe in section 4 supports these version modeling functions where appropriate. Explicitly described in that section is the environment's organization of workspaces and its organization of versions within those workspaces. Also mentioned in that section is the environment's support for resolving references to specific versions and its support for change notification and propagation. Implicit in the discussion is the environment's support for composing sets of versioned objects into configurations, and its provision for alternative groupings of versions.

Dependency maintenance

Madhavji notes that any environment supporting the orderly change of software products must provide structures for recording and reasoning about the dependencies among those software products [Madhavji 1992]. Efficient systems for keeping track of and reasoning about the dependencies and contradictions that exist among things have been developed in the field of artificial intelligence over the past twenty years. Initial research into dependency-directed backtracking during problem solving processes led to Doyle's foundational work on truth maintenance systems in the late 1970s [Doyle 1979]. Since that time much work has been published on both the formal properties and the practical applications of such systems; a recent bibliography to the literature of truth maintenance systems contains on the order of 250 entries [Martins 1991]. The environment for managing change within federated database schemas that we describe in section 4 makes use of a particular type of dependency maintenance system — the assumption-based truth maintenance system [de Kleer 1986; de Kleer and Forbus 1990] — to record and reason about the dependencies and contradictions that exist among versioned interfaces to object types.

4. AN ENVIRONMENT FOR MANAGING CHANGE

In the previous section of this paper, we gave a brief overview of established techniques in the areas of schema evolution, version modeling, and dependency maintenance that we are integrating in a solution to the problem of schema evolution for federated database systems. In this section, we describe the environment for managing change that we are developing which makes use of those techniques.

Figure 4 presents a high-level architectural view of this environment which is based on the reference model for integrated software engineering environments developed by the National Institute of Standards and Technology (NIST) and the European Computer Manufacturers Association (ECMA). Services defined in the NIST/ECMA reference model enable three forms of integration for tools within the environment [Chen and Norman 1992]:

- *data integration*, which is supported by repository and data-integration services,
- *control integration*, which is supported by process management and messaging services, and
- *presentation integration*, which is supported by user-interface services.

The foundation of this environment consists of a distributed collection of interface repositories in which component, export, and federated schemas are defined using a common object model and managed by organizations participating in the federation. Using Katz's version modeling terminology, each of these interface repositories is a *workspace* — these workspaces are organized according to the access relationships they have with one another. Interfaces defined in a workspace, A, can either inherit from or be clients of interfaces defined in another workspace, B, if and only if workspace A has access to workspace B. Overall, the set of workspaces is organized as a directed acyclic graph according to these access relationships. Although cyclical dependencies may exist among object type definitions, there is no need to permit such cycles to span workspaces. If two or more interfaces are cyclically dependent on one another, then they should be managed within the same interface repository.

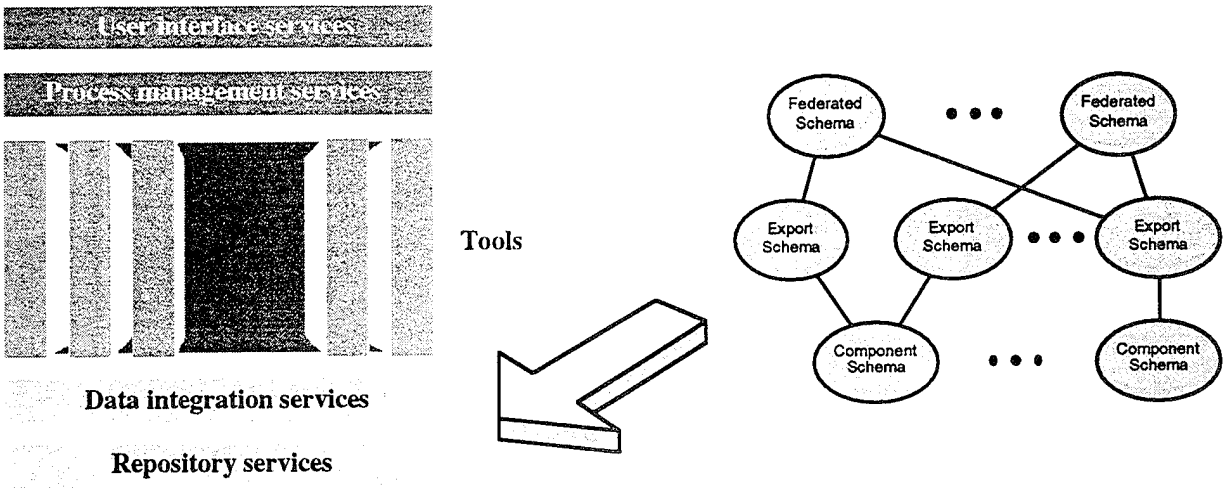


Figure 4: An Environment for Managing Change

Any environment for managing change must provide two basic types of infrastructure components: *change structures* and *dependency structures* [Madhavji 1992]. Figure 5 shows how change and dependency structures are integrated within an interface repository.

Change structures

For change structures, we will make use of version hierarchies as described by Katz. Each hierarchy records the derivation relationships among specific versions of an interface as it has changed over time. The interface version at the root of this hierarchy will be assigned version number 0 and it will have no operations, properties or constraints defined for it either locally or through inheritance. It will simply serve as a place holder for all versions of that interface that are derived as the schema evolves. Each version of an interface will each have a status associated with it: *transitional*, *working*, or *exported*. Version number 0 at the root of the interface version hierarchy will be a working version. Transitional versions may be derived from either working or exported versions and changes may be made to those transitional versions. Changes may not be made to working or exported versions.

Once a series of changes have been made to a transitional version and it has been determined to satisfy the set of schema invariants in a specific context, it may be promoted to working status. Its context (the set of interface versions it depends on) is then recorded in the interface repository's dependency maintenance system. After further evaluation of the interface (e.g. testing of its implementation) it may be exported so that it is visible to other workspaces, and versions of interfaces in those other workspaces may make use of the newly-exported interface version.

Dependency structures

For dependency structures, each interface repository in this environment will have a specific type of dependency maintenance system (known as the assumption-based truth maintenance system) which is used to record and reason about the dependencies and contradictions that arise among version of interfaces. The information we record in each dependency maintenance system as the schemas evolve can be used to perform impact analysis, change notification, and in some cases change propagation. We have also identified a method by which that information can be used to help resolve generic references to interfaces to specific version of those interfaces.

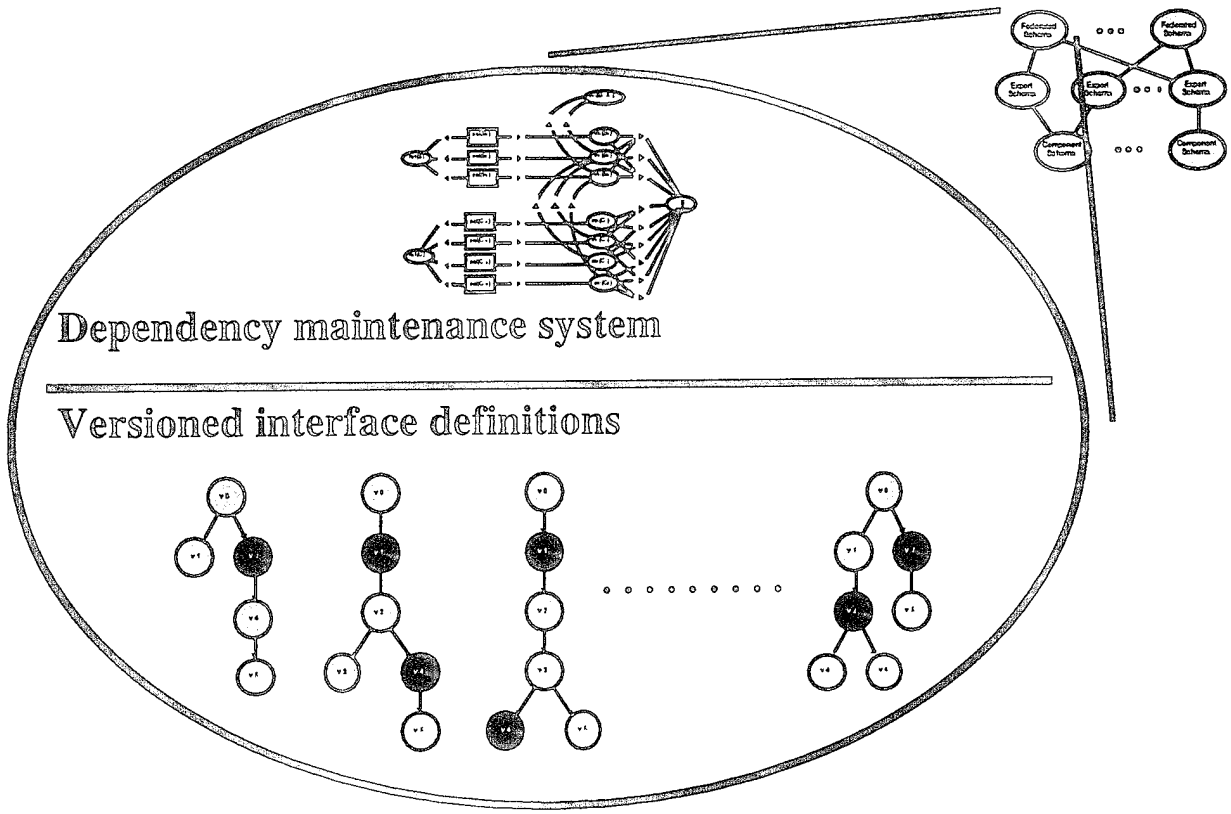


Figure 5: Integration of Change and Dependency Structures Within an Interface Repository

5. SUMMARY

In this paper we have reported on our investigation into tools and techniques that will allow organizations participating in a federation to manage change within the federated database's component, export, and federated schemas. Section 2 described the problem of schema evolution for a hypothetical federated database system that may be developed as a Leading Edge Service for GCCS. Section 3 detailed the technical approach we are pursuing in our research, and section 4 gave an overview of the environment for managing change we are developing to address the problem of schema evolution in federated database systems.

As we develop large, distributed information systems such as GCCS, we must manage the evolution of the data types on which those systems are based. A repository-based environment such as the one described here provides basic infrastructure components for managing the evolution of data type definitions expressed in a common object model. Additional tools and techniques can be incorporated into this environment to help us negotiate and plan for change among all components of such a large, distributed information system. As we proceed with this work, we will look to extend our environment with these additional capabilities.

REFERENCES

[Banerjee et al 1987a] Jay Banerjee et al. Data model issues for object-oriented applications. *ACM Transactions on Office Information Systems*, 5(1):3-26, January 1987.

[Banerjee et al 1987b] Jay Banerjee et al. Semantics and implementation of schema evolution in object-oriented databases. In *SIGMOD '87*, pages 311-322, 1987.

[Beldjilali 1995] Tarik Beldjilali. Schema evolution in the GENERAL object-oriented data model. Submitted for possible publication in the *Proceedings of TOOLS USA '95*, February 1995.

[Chen and Norman 1992] Minder Chen and Ronald J. Norman. A Framework for Integrated CASE. *IEEE Software*, pp. 18-22, March 1992.

[de Kleer 1986] Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127-162, March 1986.

[de Kleer and Forbus 1990] Johan de Kleer and Kenneth D. Forbus. Truth maintenance systems. Tutorial notes from the Eighth National Conference on Artificial Intelligence, July 1990.

[Doyle 1979] Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231-272, November 1979.

[Goldstein 1993] Theodore C. Goldstein. Supporting the Evolution of Class Definitions. In *Addendum to the Proceedings, OOPSLA '93*, pages 103-105, 1993. Workshop Report.

[Katz 1990] Randy H. Katz. Toward a unified framework for version modeling in engineering databases. *ACM Computing Surveys*, 22(4):375-408, December 1990.

[Kim and Chou 1988] W. Kim and H. T. Chou. Versions of schema for object-oriented databases. In *VLDB '88*, pages 148-159, 1988.

[Kim et al 1989] Won Kim et al. Features of the ORION object-oriented database system. In Won Kim and Frederick H. Lochovsky, editors, *Object-Oriented Concepts, Databases, and Applications*, pages 251-282. ACM Press, New York, 1989.

[Madhavji 1992] N. H. Madhavji. Environment evolution: The Prism model of changes. *IEEE Transactions on Software Engineering*, 18(5):380-392, May 1992.

[Martins 1991] J. P. Martins. The truth, the whole truth, and nothing but the truth: An indexed bibliography to the literature of truth maintenance systems. *AI Magazine*, 11(5):7-25, January 1991.

[Pons and Keller 1995] Anne Pons and Rudolf K. Keller. Resolving schema evolution by decomposition. Submitted for possible publication in the *Proceedings of TOOLS USA '95*, February 1995.

[Sheth and Larson 1990] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3), September 1990.

BIOGRAPHY

Chris Bosch is a Senior Member of the Technical Staff in the Advanced Information Technology Center at MITRE-Washington and a Ph.D. candidate in Information Technology at George Mason University. His research interests include tools and environments which promote database interoperability, support the evolutionary development of object-oriented systems, and coordinate the activities of individuals performing software systems design and development.

Contact information:

The MITRE Corporation
7525 Colshire Drive, MS Z464
McLean, VA 22102-3481
(703) 883-6902 (Office)
(703) 883-6435 (FAX)
cbosch@mitre.org

Natural Language Generator
based on
Database Modeling

Lee S Waldron, 1Lt, USAF

24 July 95

Table of Contents

		Page
	Table of Contents	2
	Abstract	3
1	Introduction	4
2	Object-Oriented Analysis	5
2.1	Object-Relationship Model	5
2.2	Object-Behavior Model	6
2.3	Modified Techniques	6
3	Original Text	6
3.1	Reasons for Non-Modeled Text	7
4	Rules of Generation	7
4.1	Generated Text	8
5	Conclusion	8
5.1	Acknowledgements	9
Appendix A:	Modifications to OSA Modeling	10
Appendix B:	Original Text	12
Appendix C:	Original Text, Text Models, and Generated Text	17
Appendix D:	Generated Text	34
Appendix E:	Algorithm for Natural Language Generation	36
Appendix F:	Rules of Generation	37

Abstract for
Natural Language Generator based on Database Modeling

Objective:

To design a program that is able to generate natural language descriptions based on information stored in a database.

Plan of Action:

1) Analyze the text from a computer science textbook.
2) Design Object-Relationship (OR) and Object-Behavior (OB) models of chosen text fragments.
3) Store into a database the designed models.
4) Study multiple ways to generate sentences, paragraphs, etc., from database models.
5) Design and implement a program that will generate natural language descriptions from information stored in a database.

Work Accomplished:

1) Developed modifications to Object-Oriented Systems Analysis (OSA) notation.
2) Designed OB and OR models of a selected text.
3) Developed rules to generate NLDs from database models.
4) Manually applied rules of generation to database information and have created an alternate text to the original.

Upcoming Work:

1) Implementation of both a graphical database and of the Natural Language Generator.
2) Design of OB model of Natural Language Generator.
3) Input OB diagram of Natural Language Generator and have the Generator compose a paper on itself.

Author:

Lt. Lee Waldron has worked within the Automated Communications System of the Standard Systems Group at Tinker AFB, OK for two years. He has done Program Management for BIDDS Management Sub-System (BMS) and lead an effort to re-write the Automated Material Management and Engineering System (AMMES) into an Access-based database. Currently, he is the Chief of Server Development on the World-Wide On Line System Replacement (WWOLS-R). He received his BS in Computer Science at Loyola University in New Orleans, La. While there, he was enrolled in ROTC next door at Tulane University (Det 320). His interest within Computer Science include Artificial Intelligence (AI) and Genetic Algorithms. Outside of that, he enjoys sports, reading, the outdoors, and physics.

Natural Language Generation

1 Introduction

The objective of this paper is to document the design of a program that is able to generate natural language descriptions (NLDs) of selected information stored in a database. In basic terms, this program will allow computers to communicate to humans in their own natural language (English for example). This will be done by analyzing the relationships between different pieces of information in a database. The kind of relationship two pieces of information have will decide what words are used by the program. The database will hold the information and the natural language generator (NLG) will determine the best way to communicate it. More accurately, the NLG will be the top layer of many translators between binary language and human language; the other translators being the code compilers and database applications. When finished, a user will be able to go into a database and select any amount of information desired. The NLG will then analyze the selected information and, using a list of sentence structure rules, translate it into plain, simple English (aka natural language descriptions). Installing an NLG, an anti-NLG (a program that reads plain English and puts it into a database), the database, and speech software, a computer will have the ability to carry on a conversation with a person. How intelligent of a conversation it will be is beyond the scope of this paper.

The needs of the NLG require a database with a graphical interface that is able to maintain relationships between Object-Relationship(OR) and Object-Behavior (OB) database models. It also requires the database to maintain multiple levels of abstraction, including relationships between abstraction instances both horizontally and vertically. In addition to these requirements, the database needs to handle relationship types developed as a result of this paper. Although such a graphical database may exist, it has eluded the author of this paper. Therefore, a new database tool is needed. The database modeling techniques of Object-Oriented Systems Analysis (OSA) as presented in Object-Oriented Systems Analysis: A Model Driven Approach by Embley, Kurtz, and Woodfield were chosen to form the basis of the new database tool. Their techniques, combined with a few modifications, became the standard method of modeling information within this paper and will be the standard within the graphical interface of the database.

This paper is based upon the status of the project at time of writing. As additional research and testing is done, the algorithm, rules, and even the scope of the paper changes. This project is nowhere near completion. In fact this paper should probably be viewed as only a draft and could probably be viewed as obsolete within a matter of weeks or months. The algorithm is changed and improved after each testing session. Although the further along the project goes, the less the algorithm changes.

Section 2 briefly describes Object-Oriented Systems Analysis with a little more detail on Object-Relationship and Object-Behavior models. Included in that section is a list of OSA techniques that were devised as a direct result of this project. The next section (section 3) explains the complete, original text and reasons why some of the text was not modeled. Within this paper, three types of text are referenced. The first is the

complete original text. This is the text as it appears within the book. The condensed original text is the portion of the complete original that was selected to be modeled. The portions that were not modeled were cut from the complete original text leaving the condensed version. Finally, the generated text is the output of the NLG after analyzing the models of the condensed original text. Section 4 goes into the heuristics of the NLD Generator; explaining the rules of generation and displaying the text from the models based on the original text and section 5 concludes the paper.

2 Object-Oriented Analysis

Object-Oriented Analysis is a database design approach which bases everything in a database on objects, just as the real-world is based on objects (i.e. cars, phones, trains). Basing a database on objects simplifies and clarifies the database by placing the emphasis of design on what an object is and how it relates to other objects rather than on how to accomplish a specific task (procedural programs). Object-Oriented Analysis not only helps to simplify and clarify the design of a database but it also helps organize all of the information concerning an object in a single location. Central location makes it easier and therefore more efficient, for the computer to locate and retrieve any information that is related to a certain object. Furthermore, the database that the NLD Generator will be using is of a graphical nature. The database is easier to understand and comprehend from a user's stand point and more descriptive and revealing from the NLD Generator's stand point when it is graphically designed. In turn, Object-Oriented Analysis encourages the designer to spend more time on the design of the objects instead of rushing past the design blindfolded. Only after a good understanding of the objects being simulated has been achieved can a programmer design an efficient and dependable database.

The Object-Oriented Analysis technique presented in Object-Oriented Systems Analysis: A Model-Driven Approach, is called Object-Oriented Systems Analysis (OSA). In OSA, there are two principle types of design techniques or models, The first is Object-Relationship (OR) models. OR models display objects and their relationships to other objects. The second is Object-Behavior (OB) models. OB models, on the other hand, illustrate which circumstances cause objects to perform which actions and when.

2.1 Object-Relationship Model

An Object-Relationship Model describes objects and their relationships to other objects. An object can be any item; a car, a person, a key, etc. A relationship is the connection between two or more objects which describes how they interact between themselves. One example is 'Dave attends Loyola'. Both 'Dave' and 'Loyola' are objects. 'Attends' is the relationship. 'Attends' describes how 'Dave' and 'Loyola' interact with each other. An object class represents a group of objects that have similar characteristics and behavior. 'Dave' and 'Steven' (objects) are examples of student (object class). Relationships with a common meaning are also grouped together. They are called relationship sets. Let's say that 'Dave (object) attends (relationship) Loyola' (object). Let's also say that 'Steven (object) attends (relationship) Loyola'. These two relationships are alike in that they both connect a student (the object class in which both

'Dave' and 'Steven' are instances of) to a University (the object class that 'Loyola' is an instance of). From these two relationships, we can create a set called attends. Our model would then show: student(object class) attends (relationship) university (object class). An object class is represented by a rectangle with the name of the object within the rectangle. A relationship set is represented as a line connecting the participating object classes together with the name of the relationship set near the line.

2.2 Object-Behavior Model

An Object-Behavior (OB) model describes how an object acts and reacts according to different stimuli. Whereas the Object-Relationship model shows the permanent conditions of an object, the Object-Behavior model portrays the temporary aspects. An object will traverse through different states of the OB model in the manner determined by which conditions are met and when. For example, Dave goes to play pool on Thursday afternoons. But if Dave has a test the next day, he will study before he plays pool. Dave is the object and playing pool is the activity that the object engages in when the condition day = Thursday is met. 'Dave' starts out in the state of 'idle'. When he plays pool, he moves to the state of 'enjoyment'. If the condition has a test the next day is met then the object will study. At that point 'Dave' moves into the state 'studious'. If both conditions are met then the condition with the highest priority is chosen. In this case, studying has a higher priority than playing pool. As you can tell, the condition 'Dave is a Senior' has not been met.

An object (or object class) will be the subject of an OB model. The OB model will show the state of an object and the behaviors that the object exhibits if certain conditions were met. A state is the status of the object at a particular moment. A car can be idle. A person can be sitting. A computer can be on. All of these are states of different objects. A state is shown as a circle with the name of the state within it. A behavior is shown as a box with the trigger in the top section and the reaction in the lower section of the box. The trigger is the condition that must be met in order for the object to perform the actions listed in the reaction section.

2.3 Modified Techniques

The nature of this program demands more from the design of a database than most applications. Some of these demands were not supported by the structure of OSA techniques. Because of this, it was found to be necessary to create my own modifications to the techniques presented in the Emberly text. Appendix A shows the list of modifications that were incorporated in later models. Next to the symbol is the description of what the symbol is for and why it was needed.

3 Original Text

To begin the project, database had to be created from which NLDs could later be generated from. A portion of Essentials of Computing by Carmon was chosen to be the test data. This book was chosen because of its straight-forwardness in its wording. It

provided information that was written in plain, simple English. There was no flowery language nor complicated sentence structures. Eventually the NLD Generator should be able to mimic such text but to start with it simpler is better.

The database was created (or more properly, simulated having been created) using this text. Appendix B contains the text used. Those parts that are underlined show the sections that were actually used in creating the database model. Those sentences that are not underlined, have a number in parenthesis immediately following the sentence. This number corresponds to the reason why it was not used in the database creation process (covered n Section 3.1). Thirdly, there re some sentences that are italicized. These sentences were diagrammed on their own as an example (instance) of the more abstract model created from the underlined text.

3.1 Reasons for Non-Modeled Text

It was necessary to identify those sections that could not be diagrammed and the reason for that inability. This is so that in the future such inabilities can be remedied. Some of these reasons are as follows.

1. This sentence establishes the direction in which the paper will describe the information. The program will decide the direction of description based on it's rules. There is no need to diagram this.

2. A satisfactory notation has not yet been devised that describes an object class and an object at the same time.

3. This information either has been previously stated or is stated in a better form later in the text.

4. There are some verb tenses that do not yet have an equivalent notation.

5. There are certain conditions upon the verb that do not have equivalent notation.

6. This makes reference to items not included in the original text.

7. This is a Webster definition of the word. It doesn't need to be modeled

4 Rules of Generation

After creating the database model of the text, a method of describing the information was needed. This was acquired by studying the structure and meaning of the database model. Initially, attention was given to developing rules for single relationships (object - relationship - object). Basic sentences were born from this. (subject verb direct object). Next, conditions or characteristics of the relationship were analyzed. This provided verb describing adjectives, adverbs, and clauses. Later, different levels of abstraction were dissected. From this arose different threads of conversation. Thought

was given to how to handle this. A preliminary decision was arrived at but, like all of the other rules, will probably change as more examples are analyzed and tested. Finally, the relationships between OR and OB models were looked at. This was perhaps the hardest to develop a plan for. To put it simply, one type of model will be handled before moving onto the other. This eliminated the need to keep track of different threads. The analysis is what spawned the majority of the rules listed in Appendix F. When an initial generation of NLDs was attempted, a very rough set of sentences that did not flow well and at times didn't make sense was produced. Another large portion of the rules resulted from this initial generation. It was discovered that most sentences can be broken into the following sentence structure or a close facsimile of it.

<object> <relationship> <object>

Although this seems very elementary, an attempt was made to conduct the analysis with no assumptions being made prior to the analysis in order to keep from the analysis becoming skewed. One of the discoveries was that the type of verb determines the modeling technique needed for that sentence. If the verb is a verb of being; an OR model is needed. If the verb is a verb of doing (action); an OB model is required. These rules are included within the rules listed in the appendix.

4.1 Generation of Text

After the condensed original text was modeled, NLD generation was simulated using the developed NLDG algorithm (shown in Appendix E). The algorithm was developed by assigning a priority to each rule of generation and rule of traversing. The applicable rule with the highest priority is used first in describing the information. If more than one rule is applicable, all are used unless there is a conflict between rules. In the event of a conflict, the rule with the highest priority is used. By ordering the rules of traversement and rules of generation, a procedural language pseudo-code was able to be developed. It was this pseudo-code, or algorithm, that was used in the simulation of text generation. Of course, this algorithm is not a finished product. In actuality, it changes quite often due to more research and testing. Appendix C shows the condensed original text, the models created from this text, and the text generated from the models using the NLDG algorithm. The numbers that follow each sentence of the generated text corresponds to the rules used to create that sentence. Appendix D is the concatenation of all of the generated text. This is the final product.

5 Conclusion

The next step in my research process is the design and implementation of a graphical database that incorporates the necessary OSA techniques. After that design and implementation of the NLDG program is needed. The goal is to include all of the information within this paper within the database and have the NLDG write a similar paper on itself.

This paper is in no way comprehensive in the description of all the ways to describe all forms of English. I have covered only a limited amount of English tenses,

grammar, etc. English is much too broad of a language to be dissected in such a short amount of time.

5.1 Acknowledgments

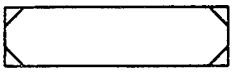
I must first and foremost thank Dr. Bogdan Czejdo, professor of Computer Science at Loyola University, for his guidance and help. Also for that occasional (or not so occasional) kick-in-the-pants. With out his knowledge, this paper would still only be an idea. Also, David O'Leary for being the sounding board for all of this.

Appendix A

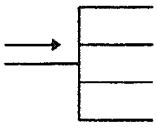
Modifications to OSA Modeling



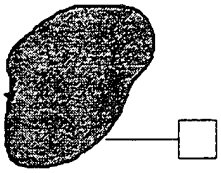
= Signifies that the relationship is a high level notation for a lower level diagram



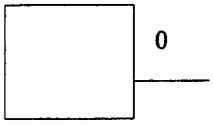
= Signifies that the object is the root of the diagram.



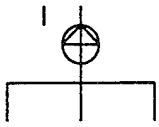
= Signifies a multi-object binary relationship.



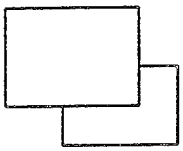
= Signifies an exploded view of a high level object with objects outside of the exploded view that were inappropriate to show on the high level view.



= Signifies that the object never participates in the said relationship.



= Signifies that all of the components of the Aggregate class are shown.



= Signifies that the object (box in fore view) has an instance (box attached to the object) that is of relevance to the subject.



= Signifies that the object has more than one name that it can be referred to by. The more common name is placed in the top section of the box.

Appendix B

Original Text

Data: Getting Organized

To be processed by the computer, data-represented by characters- is organized into fields, records, files, and sometimes databases. We will start with the smallest element, the character. (A)

¥ A character is a letter, number, or special character (such as $, ? or *). One or more related characters constitute a field.

¥ A field contains a set of related characters. *For example, suppose a health club is making address labels for a mailing. (B) For each person, it has a member-number field, a name field, a street-address field, a city field, a state field, a zip-code field, and a phone-number field. (B)*

¥ A record is a collection of related fields. *Thus, on the health-club list, o ne person s member-number, name address, city, state, zip code, and phone number constitute a record. (B) (The fields are considered related because they are for the same person.) (B)*

¥ A file is a collection of related records. *All the member records for the health club compose a membership file. (B)*

¥ A database is a collection of interrelated files stored together. (However, a file is not necessarily part of a database; menu files exist independently.) In a database, specific data items can be retrieved for various applications. *For instance, if the health club is opening a new outlet, it can pull out the names and addresses of all the people with specific zip codes that are near the new club. (B) The club can then send a special announcement about opening day to those people. (B)*

Processing Data into Information: A User Perspective

There are several methods of processing data in a large computer network of mainframe system. The two main methods are batch processing (processing data transactions in groups) and transaction processing (processing the transactions one at a time as they occur). A combination of these two techniques may also be used. We will now look at these methods and give examples of their use. (A)

Batch Processing

Batch processing is a technique in which transactions are collected into groups, or batches, to be processed. *Let us suppose that we are going to update the health club address label file. (B) The master file, a semi-permanent set of records, is, in this case, the records of all members of the health club, including their names, addresses, and so forth. (B)*

All changes to be made to the master file are compiled on a separate transaction file. Such changes can be of the following types:

- ¥ Additions are transactions to create new master records *for new names added. (B) If Sally Kelley is joining the club, a transaction containing the fields for Ms. Kelley-including member number, name, address, and so forth- will be prepared to add the new member record to the file. (B)*

- ¥ Deletions are transactions with instructions to remove master records *of people who have resigned form the health club. (B) For example, if Ha Dao resigns from the club, a transaction is prepared to remove her record from the file. (B)*

- ¥ Revisions are transactions to change fields *such as street addresses or phone numbers on the master records. (B) For example, if Benson Porter changes his address and phone number, a transaction is prepared to reflect these changes on his record in the master file. (B)*

At regular intervals, perhaps monthly in this example, the master file is updated with the changes called for on the separate transaction file. The result is a new, up-to-date master file. *The new file in this example has a new record for Sally Kelley, no longer has a record for Ha Dao, and has a changed record for Benson Porter (B).*

An advantage of batch processing is that it is usually less expensive than other types of processing because it is more efficient: A group of records is processed at the same time. (C) A disadvantage of batch processing is that anyone interested in the outcome- customers or business users- has to wait. (C) It does not matter that you want to know what the gasoline bill for your car is now; you have to wait until the end of the month, when all your credit card has purchases are processed together with those of other customers. (B) Batch processing cannot give you a quick response to your question.

Transaction Processing

Transaction processing is a technique of processing transactions one at a time in random order - that is, in any order they occur. Transaction processing is handy for anyone who needs an immediate update or feedback from the computer: a contractor who needs to check a supplier s rates; an airline clerk making a res-

ervation; a retailer who wants to confirm product inventory; and many, many others. (B) In fact, transaction processing has become a staple in all kinds of service industries in which speedy service is a must.

Transaction processing is real-time processing. Real-time processing can obtain data from the computer system in time to affect the activity at hand. In other words, a transaction is processed fast enough for the results to come back and be acted upon right away. (C) For example, a teller at a bank (or you at an automated teller machine) can find out immediately what your bank balance is. (C) You can then decide right away how much money you can afford to withdraw. (C) For processing to be real-time, it must also be on-line - that is, the user s terminal must be directly connected to the computer.

The great leap forward that transaction processing represents was made possible by the development of magnetic disk as a means of storing data. With magnetic tape it is not efficient to go directly to the particular record you are looking for - the tape might have to be advanced several feet first. (A) However, with disk you can go directly to one particular record. The invention of magnetic disk meant that data processing is more likely to be interactive, as is possible with the personal computer. (E) The user can communicate directly with the computer, maintaining a dialogue back and forth. The direct access to data on disk dramatically increases the use of interactive computing.

There are several advantages to transaction processing. The first is that you do not need to wait. For instance, a department store salesclerk using a point-of-sale terminal can key in a customer s charge-card number and a code that asks the computer Is this charge card acceptable? and get an immediate reply. (B) Immediacy is a distinct benefit, since everyone expects fast service these days. (C) Second, the process permits continual updating of a customer s record. Thus the salesclerk can not only verify your credit but also record the sale in the computer, and you will eventually be billed through the computerized billing process. (B)

Figure 5-2 provides an example of transaction processing, in which a patient submits a prescription. (F)

Batch and Transaction Processing: Complementary

Numerous computer systems combine the best features of both of these methods of processing. (C) A bank, for instance, may record your withdrawal transaction during the day at the teller window whenever you demand your cash. (B) However, the deposit that you leave in an envelope in an instant deposit drop may be recorded during the night by means of batch processing. (B)

Another common example of both batch and transaction processing is in retail sales. (B) Using point-of-sale terminals, inventory data is captured as sales are made; this data is processed later in batches to produce inventory reports. (B)

Storage Media

As we have mentioned, two primary media for storing data are magnetic

tape and magnetic disk. Since these media have been the staples of the computer industry for three decades, we will begin with them. (A)

Magnetic Tape Storage

Magnetic tape looks like the tape used in home tape recorders - plastic Mylar tape, usually 1/2 inch wide and wound on a 10 1/2 inch diameter reel. The tape has an iron-oxide coating that can be magnetized. Data is stored as extremely small magnetized spots, which can then be read by a tape unit into the computer s main storage.

Figure 5-4a shows a magnetic tape unit that is part of a large computer system. (F) The purpose of the unit is to write and to read - that is, to record data on and retrieve data from - magnetic tape. This is done by a read/write head. Reading is done by an electromagnet that senses the magnetized areas on the tape and converts them into electrical impulses, which are sent to the processor. The reverse is called writing. Before the machine writes on the tape, the erase head erases any previously recorded data.

Records are stored on tape sequentially - that is, in order by some identifier such as a social security number.

Magnetic Disk Storage

Magnetic disk storage is another common form of secondary storage. A hard magnetic disk, of hard disk, is a metal platter coated with magnetic oxide that looks something like a large brown compact disk. Hard disks come in a variety of sizes; 14, 5 1/4, 3 1/2 inches are typical diameters. Several disks of the same size are assembled together in a disk pack. A disk pack looks like a stack of stereo records, except that daylight can be seen between the disks. There are different types of disk packs, with the number of platters varying by model. Each disk has a top and bottom surface on which to record data.

Another form of magnetic disk storage is the diskette, which is a round piece of plastic coated with magnetic oxide. Diskettes and small hard disk are used with personal computers. We will discuss secondary storage for personal computers later in this chapter, but keep in mind that the principles of disk storage discussed here also apply to disk storage for personal computers. (A)

How Data is Stored on a Disk

As Figure 5-6 shows, the surface of each disk has tracks on it. Data is recorded as magnetic spots on the tracks. The number of tracks per surface varies with the particular type of disk. A track on a disk is a closed circle - any point on a particular track is always the same distance from the center. (G) All tracks on one disk are concentric - that is, they are circles with the same center. (G)

The same amount of data is stored on every track, from outermost (track 000) to innermost (track 399 of a 400-track disk), and it takes the same amount of

time to read the data on the outer track as on the inner, even though the outer track moves faster.

A magnetic disk is a direct-access storage device (DASD). With such a device you can go directly to the record you want. (C) With tape storage, on the other hand, you must read all preceding records in the file until you come to the desired record. Records can be stored either sequentially or randomly (in whatever order the records occur) on a direct-access storage device.

Appendix C

Modeled Text with Original Text and Generated Text

Original Text:

Data: Getting Organized {Fig. 1}

To be processed by the computer, data-represented by characters- is organized into fields, records, files, and sometimes databases.

¥ A character is a letter, number, or special character (such as $, ? or *). One or more related characters constitute a field.

¥ A field contains a set of related characters.

¥ A record is a collection of related fields.

¥ A file is a collection of related records.

¥ A database is a collection of interrelated files stored together. (However, a file is not necessarily part of a database; menu files exist independently.) In a database, specific data items can be retrieved for various applications.

Processing Data into Information: A User Perspective

There are several methods of processing data in a large computer network of mainframe system. The two main methods are batch processing (processing data

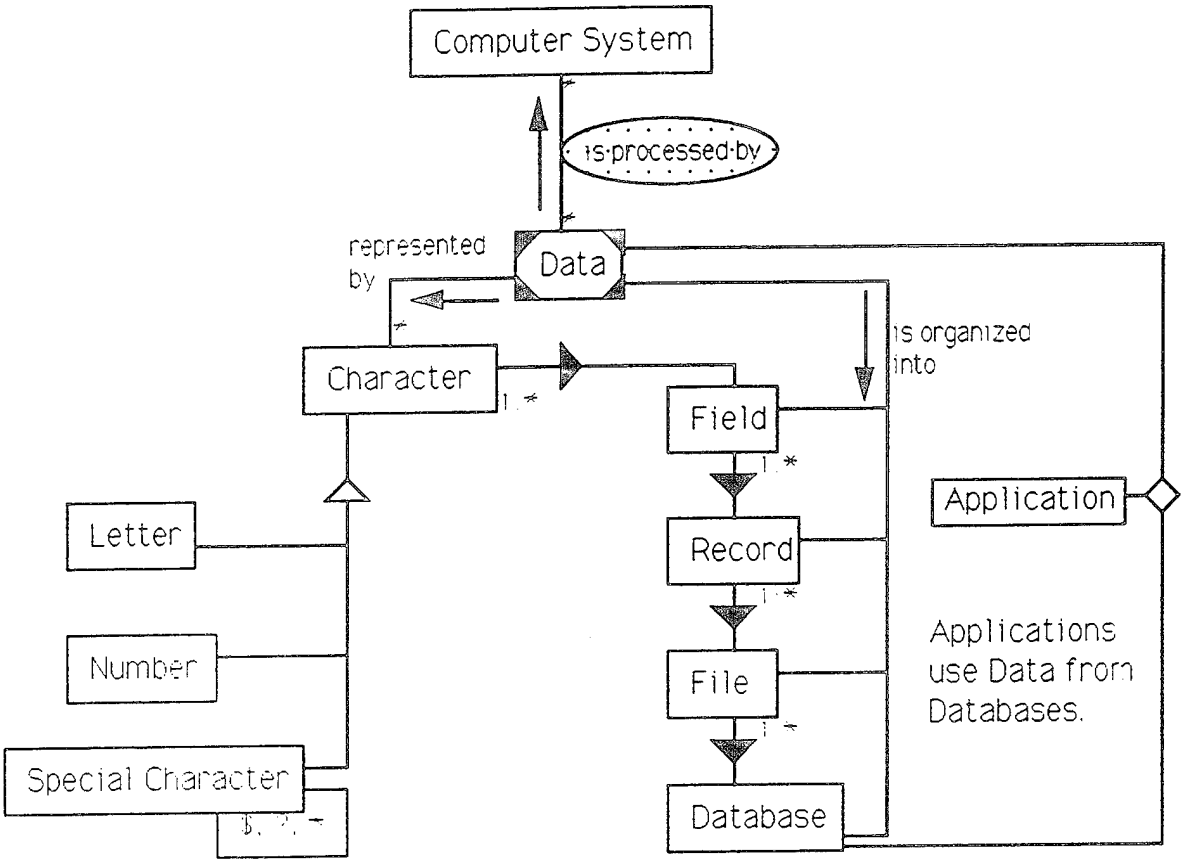


Fig. 1-1 Data - Getting Organized

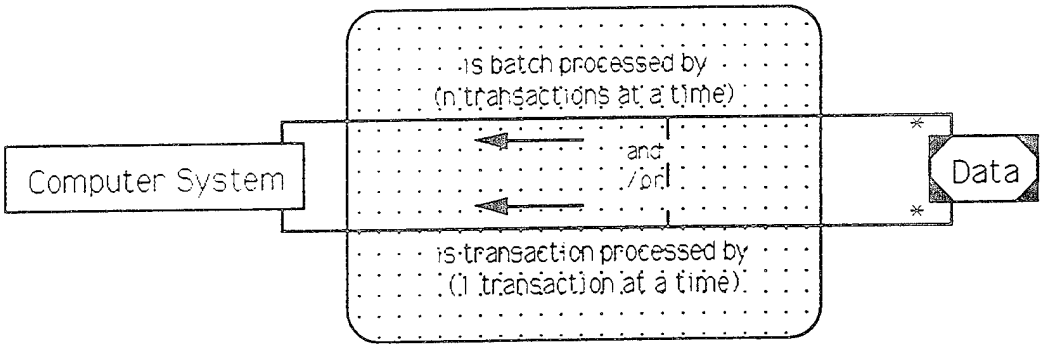


Fig. 1-2 Low level of 'Is Processed By' relationship

GENERATED TEXT:

Data is processed by Computer Systems (1,7,9,18). By this we mean Data is batch processed by (n transactions at a time) and/or is transaction processed by (1 transaction at a time) by Computer Systems (2,7,9,13,18). Data is represented by Characters (5,9,18). A Letter, a Number, and a Special Character are all types of Characters (17). Examples of a Special Character are $, ?, and *.(5) One or More Characters compose a Field (9,18). One of more fields compose a record (9,18). One or more records compose a file (9,18). One or more files compose a database (9,18). Data is organized into fields, records, files, and databases (14,15,18). Applications use Data from Databases.(14,19)

Original Text:

Batch Processing

Batch processing is a technique in which transactions are collected into groups, or batches, to be processed.

All changes to be made to the master file are compiled on a separate transaction file. Such changes can be of the following types:

- Additions are transactions to create new master records...
- Deletions are transactions with instructions to remove master records...
- Revisions are transactions to change fields...

An advantage of batch processing is that it is usually less expensive than other types of processing because it is more efficient. Batch processing cannot give you a quick response to your question.

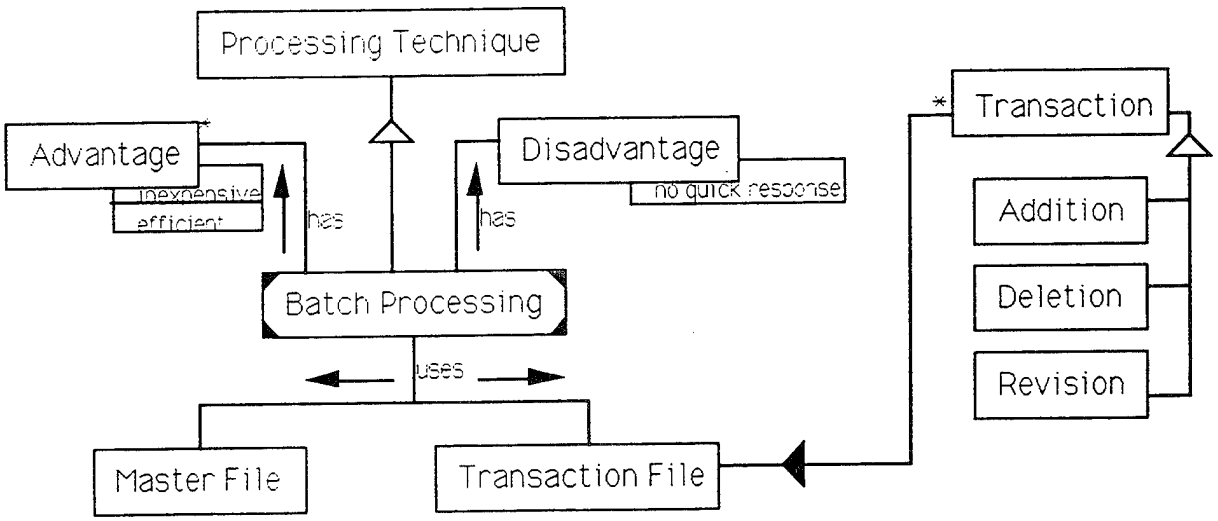


Fig. 2-1 Batch Processing

Generated Text:

Batch Processing is a Processing Technique (7,17). Batch Processing has Advantages.(9,18) Examples of Advantages are inexpensive and efficient (5). Batch Processing has Disadvantages (9,18). An example of a Disadvantage is no quick respense (5) Batch Processing uses a Master File and a Transaction File (15,18). Transactions compose a Transaction File (9,20). Addition, Deletion, and Revision are types of Transactions (17).

Original Text:

Batch Processor

Batch processing is a technique in which transactions are collected into groups, or batches, to be processed.

All changes to be made to the master file are compiled on a separate transaction file. Such changes can be of the following types:

- Additions are transactions to create new master records...
- Deletions are transactions with instructions to remove master records...
- Revisions are transactions to change fields...

At regular intervals, perhaps monthly in this example, the master file is updated with the changes called for on the separate transaction file. The result is a new, up-to-date master file.

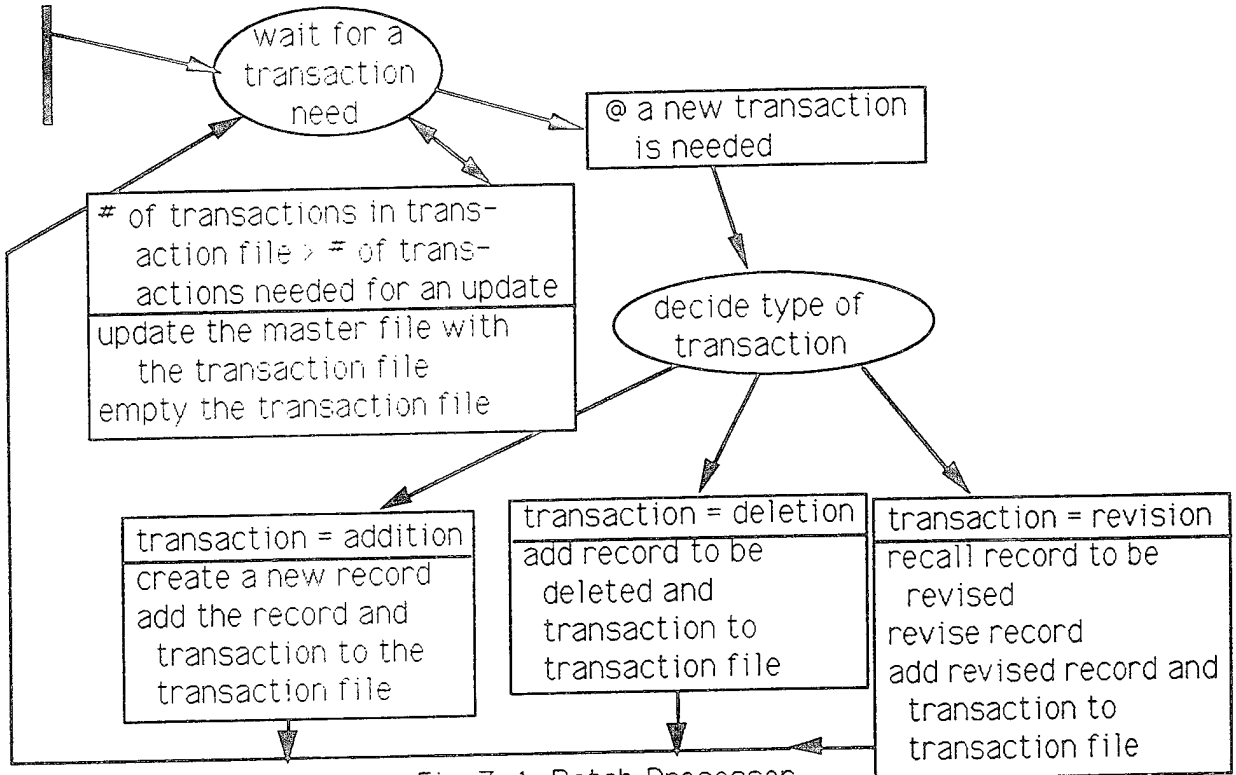


Fig. 3-1 Batch Processor

Generated Text:

In the case of a Batch Processor, it waits for a transaction (26). If the number of transactions is greater than the number needed for an update, then update the master file with the transaction file (3,23,27,). When a new transaction is needed, create a new transaction (3,23). If the transaction is addition, then create a new record and add the record and transaction to the transaction file (3,23,27). If the transaction is deletion, add the record to be deleted and the transaction to the transaction file (3,23,27). If the transaction is revision, them recall the record to be revised (3,23,27). Then revise the record and add the revised record and the transaction to the transaction file (3,23,27).

Generated Text

Transaction Processing

Transaction processing is a technique of processing transactions one at a time in random order - that is, in any order they occur. Transaction processing is handy for anyone who needs an immediate update or feedback from the computer.

Transaction processing is real-time processing. Real-time processing can obtain data from the computer system in time to affect the activity at hand.

The great leap forward that transaction processing represents was made possible by the development of magnetic disk as a means of storing data.

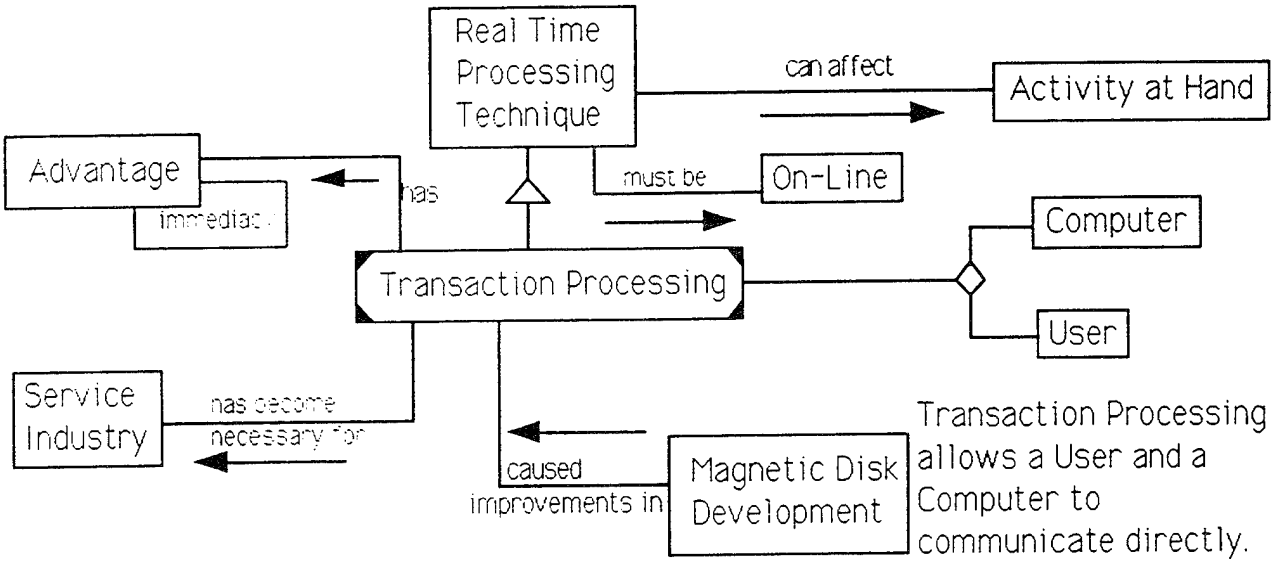


Fig. 4-1 Transaction Processing

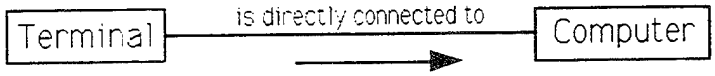


Fig. 4-2 Low-Level of 'On-Line' Object

Generated Text

Transaction Processing is a Real-Time Processing Technique (7,17). It can affest the Activity at Hand (4,18). A Real Time Processing Technique must be On-Line (18). By this we mean, a Terminal is directly connected to a Computer (2,13,18). Transaction Processing has Advantages (8,18). An example of an Advantage is immediacy (11). Transaction Processing has become necessary for Service Industry (18). Magnetic Disk Development caused Improvements in Transaction Processing (8,18). Transaction Processing allows a User and a Computer to communicate directly (19).

Original Text:

Transaction Processor

Transaction processing is a technique of processing transactions one at a time in random order - that is, in any order they occur.

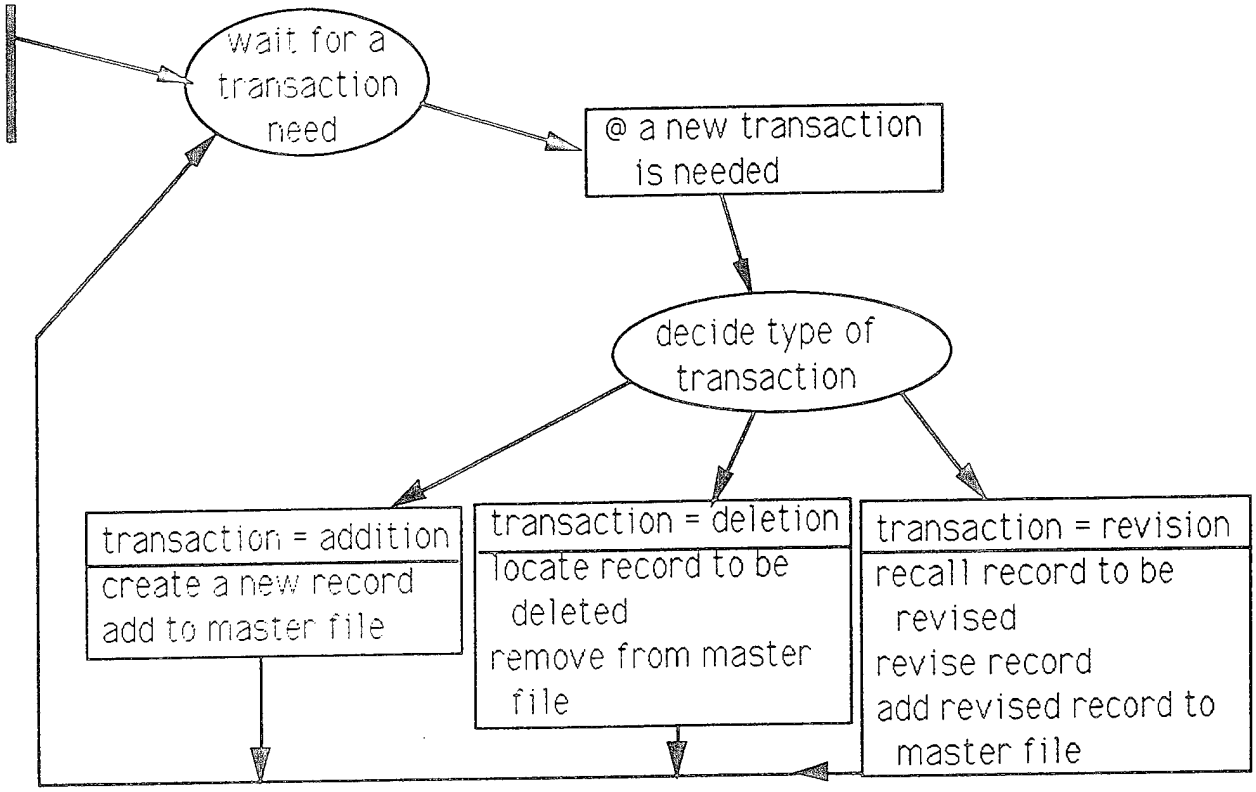


Fig. 5-1 Transaction Processor

Generated Text:

A Transaction Processor will wait for a transaction need (26). When a new transaction is needed, decide the type of transaction (23). If the transaction is an addition, then create a new record and add it to the master file (23,27). If the transaction is a deletion, locate the record to be deleted and remove it from the master file (23,27). If the transaction is a revision, recall the record to be revised (23,27). Then revise the record and add the revised version to the master file (22).

Original Text (Fig. 6):

Storage Media

As we have mentioned, two primary media for storing data are magnetic tape and magnetic disk.

Magnetic Tape Storage

Magnetic tape looks like the tape used in home tape recorders - plastic Mylar tape, usually 1/2 inch wide and wound on a 10 1/2 inch diameter reel. The tape has an iron-oxide coating that can be magnetized. Data is stored as extremely small magnetized spots, which can then be read by a tape unit into the computer s main storage.

The purpose of the unit is to write and to read - that is, to record data on and retrieve data from - magnetic tape. This is done by a read/write head. Reading is done by an electromagnet that senses the magnetized areas on the tape and converts them into electrical impulses, which are sent to the processor. The reverse is called writing. Before the machine writes on the tape, the erase head erases any previously recorded data.

Records are stored on tape sequentially - that is, in order by some identifier such as a social security number.

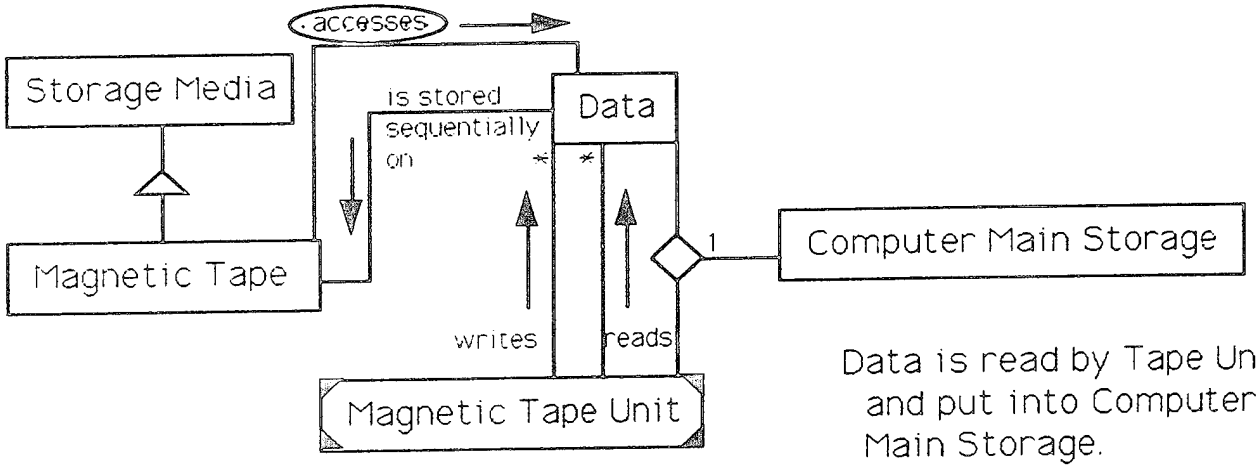


Fig. 6-1 Storage Media

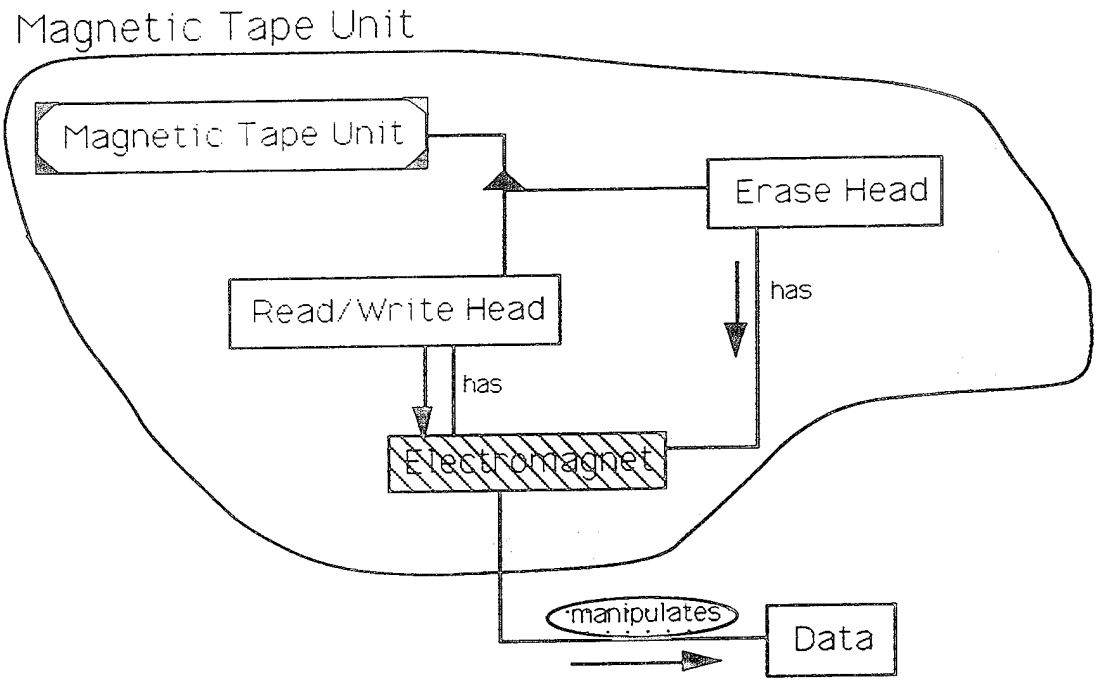


Fig. 6-2 Low-Level of 'Magnetic Tape Unit' Object

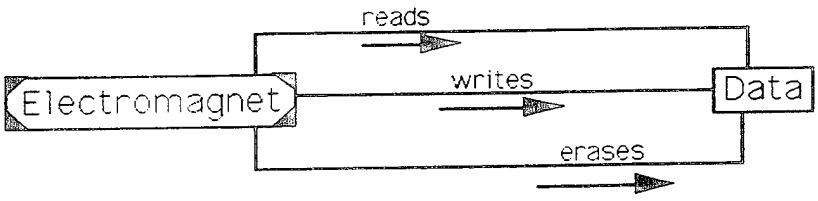


Fig. 6-3 Low-Level of 'Manipulates' Relationship

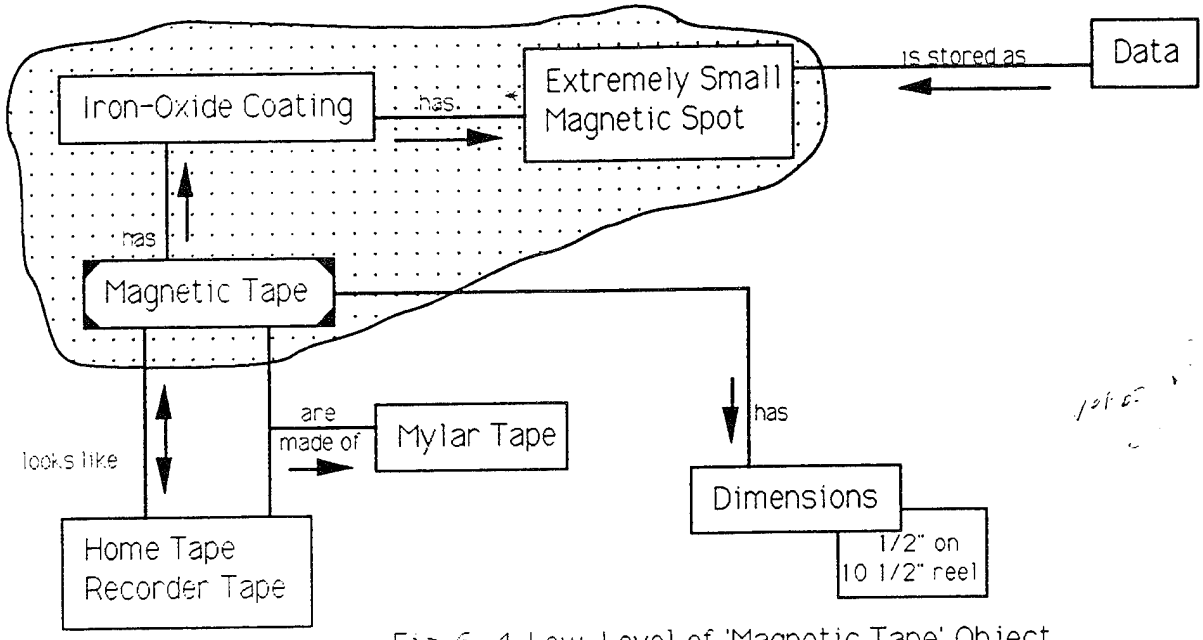


Fig. 6-4 Low-Level of 'Magnetic Tape' Object

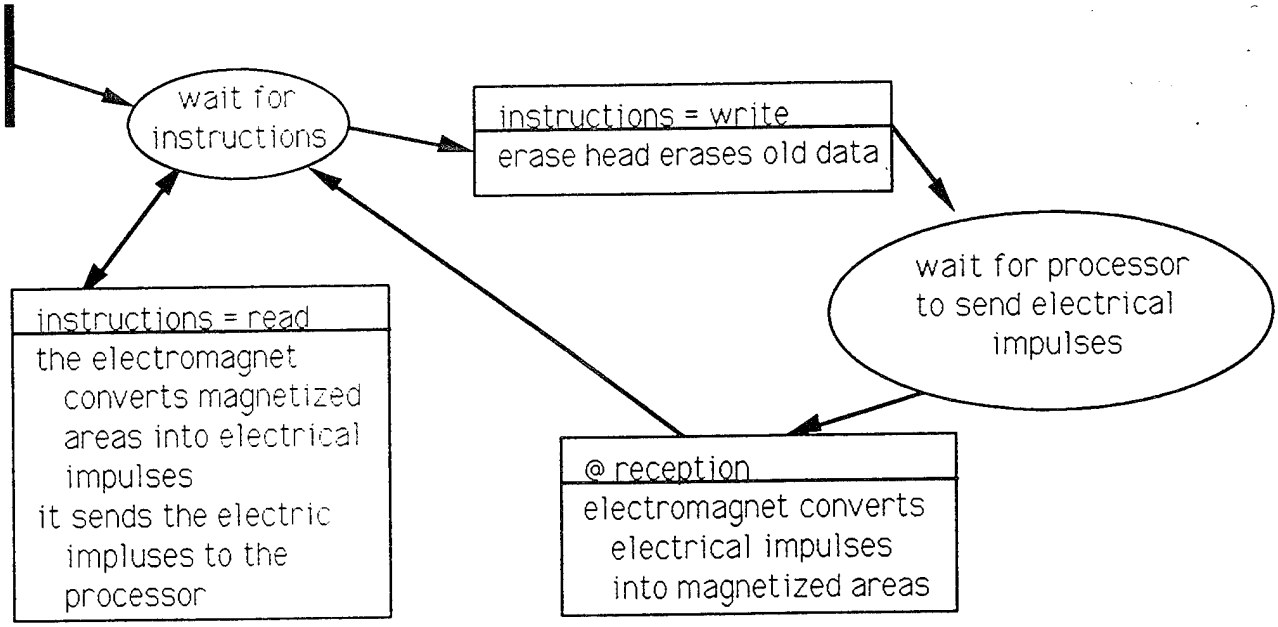


Fig. 6-5 Low-Level of 'Manipulates' Relationship

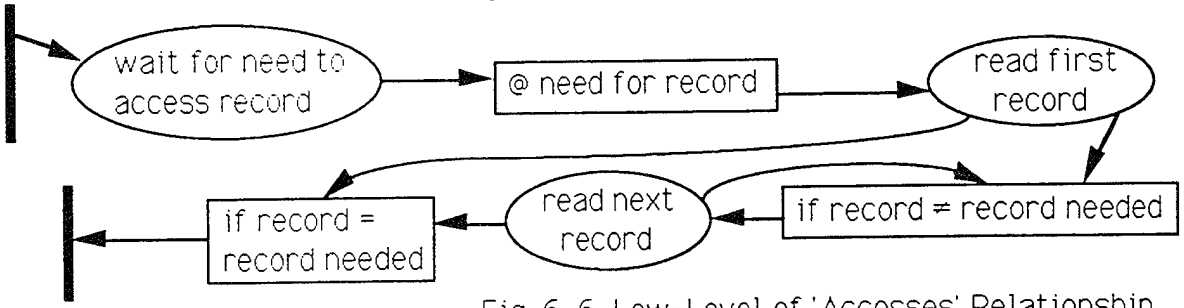


Fig. 6-6 Low-Level of 'Accesses' Relationship

Generated Text (Fig. 6):

A Magnetic Tape Unit reads and writes Data (10,12). Magnetic Tape Unit is composed of a Read/Write Head and an Erase Head (28). Both have an electromagnet (4,26). An Electromagnet manipulates Data (26). By this we mean, an Electromagnet reads, writes, and erases Data (12,16). Data is stored sequentially on Magnetic Tape (26). Magnetic Tape looks like Home Tape Recorder Tape (26). Both are made of Mylar Tape (4,26). Magnetic Tape has Dimensions (26). An example of Dimensions is 1/2" on 10 1/2" reel (14). Magnetic Tape has an Iron-Oxide Coating (26). An Iron-Oxide Coating has Extremely Small Magnetic Spots (26). Data is stored as Extremely Small Magnetic Spots (26). Magnetic Tape is a Storage Media (25). Magnetic Tape accesses Data (26). By this we mean, it waits for a need to access a record (16,44). When a need for a record is present, read the first record (41). If the record read is the record needed then it is finished (41). If the record read is not the record needed then read the next record (41). Data is read by a Tape Unit and put into Computer Main Storage (27). A Magnetic Tape Unit will wait for instructions (44). If the instructions are to read, then the electromagnet converts magnetized areas into electrical impulses (41). Then it sends the electric impulses to the processor (41). Then the Magnetic Tape Unit returns to wait for instructions (41). If the instructions are to write, then the erase head erases the old data (41,45). The Magnetic Tape Unit will then wait for the processor to send the electrical impulses (41). At reception, the electromagnet converts the electrical impulses into magnetized areas (41). The Magnetic Tape Unit will then wait for instructions (40).

Original Text (Fig. 7):

Magnetic Disk Storage

Magnetic disk storage is another common form of secondary storage. A hard magnetic disk, of hard disk, is a metal platter coated with magnetic oxide that looks something like a large brown compact disk. Hard disks come in a variety of sizes; 14, 5 1/4, 3 1/2 inches are typical diameters. Several disks of the same size are assembled together in a disk pack. Each disk has a top and bottom surface on which to record data.

Another form of magnetic disk storage is the diskette...

How Data is Stored on a Disk

...the surface of each disk has tracks on it. A track on a disk is a closed circle... All tracks on one disk are concentric...

A magnetic disk is a direct-access storage device (DASD). Records can be stored either sequentially or randomly (in whatever order the records occur) on a direct-access storage device.

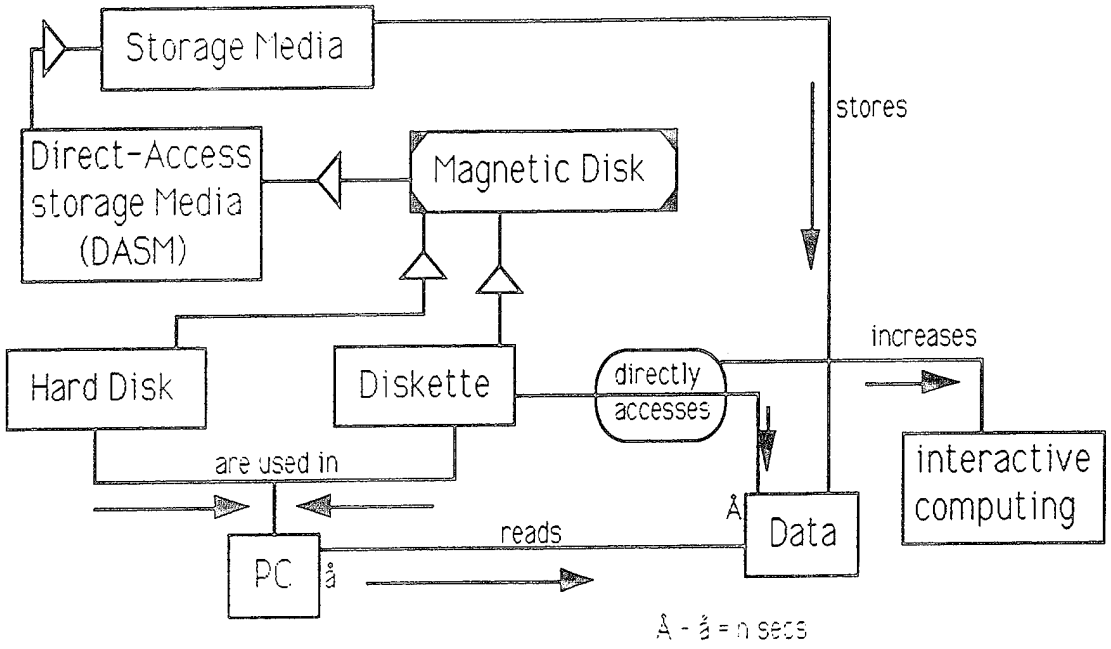


Fig. 7-1 Magnetic Disk



Fig. 7-2 Low-Level of Direct-Access Storage Device

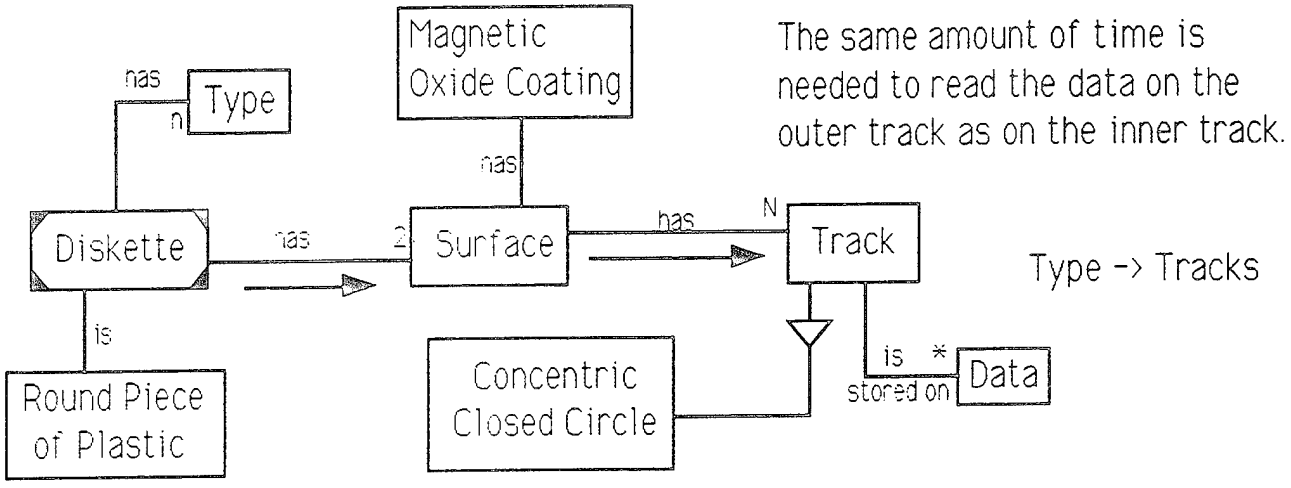


Fig. 7-3 Low-Level of Diskette

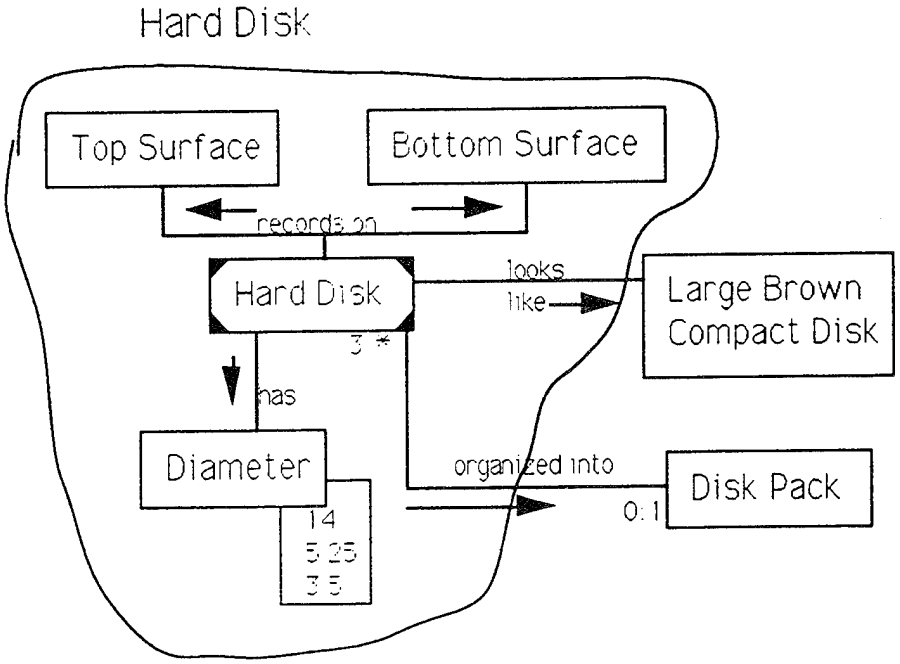


Fig. 7-4 Low-Level of Hard Disk

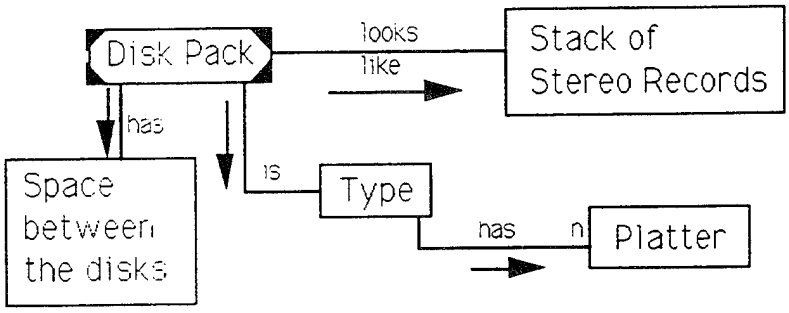


Fig. 7-5 Low-Level of Disk Pack

Generated Text (Fig. 7):

A Magnetic Disk is a Direct-Access Storage Media (DASM) (10,25). A DASM sequentially and randomly stores Records (12). DASM is a Storage Media (25). Storage Media stores Data (26). Hard Disks and Diskettes are Magnetic Disks (18). A Hard Disk looks like a Large Brown Compact Disk (26). It records on a Top Surface and a Bottom Surface (4,18,26). A Hard Disk has a Diameter (26). Examples of Diameter are 14 inch, 5.25 inch and 3.5 inch (14). Three or more Hard Disks are organized into a Disk Pack (11,26). A Disk Pack looks like a stack of Stereo Records (26). It has space between the disks (4,26). A Disk Pack is a Type (26). A Type has N Platters (26). A Diskette has two Surfaces (11,26). A Surface has N tracks (11,26). A Track is a Concentric Closed Circle (25). Data is stored on a Track (26). A Surface has a Magnetic Oxide Coating (26). A Diskette is a Round Piece of Plastic (25). It has N Types (4,26). The same amount of time is needed to read the data on the outer track as on the inner track (6). The Type determines the Tracks (20). Hard Disks and Diskettes are used in a PC (18,26). A PC reads Data within N seconds (15,26). A Diskette directly accesses Data (26). The Direct Access increases Interactive Computing (26).

Original Text (Fig. E):

For example, suppose a health club is making address labels for a mailing. For each person it has a member-number field, a name field, a street-address field, a city field, a state field, a zip-code field, and a phone-number field.

Thus, on the health-club list, one person s member-number, name address, city, state, zip code, and phone number constitute a record. (The fields are considered related because they are for the same person.)

All the member records for the health club compose a membership file.

For instance, if the health club is opening a new outlet, it can pull out the names and addresses of all the people with specific zip codes that are near the new club. The club can then send a special announcement about opening day to those people.

Let us suppose that we are going to update the health club address label file. The master file, a semi-permanent set of records, is, in this case, the records of all members of the health club, including their names, addresses, and so forth.

If Sally Kelley is joining the club, a transaction containing the fields for Ms. Kelley-including member number, name, address, and so forth- will be prepared to add the new member record to the file.

For example, if Ha Dao resigns from the club, a transaction is prepared to remove her record from the file.

For example, if Benson Porter changes his address and phone number, a transaction is prepared to reflect these changes on his record in the master file.

The new file in this example has a new record for Sally Kelley, no longer has a record for Ha Dao, and has a changed record for Benson Porter.

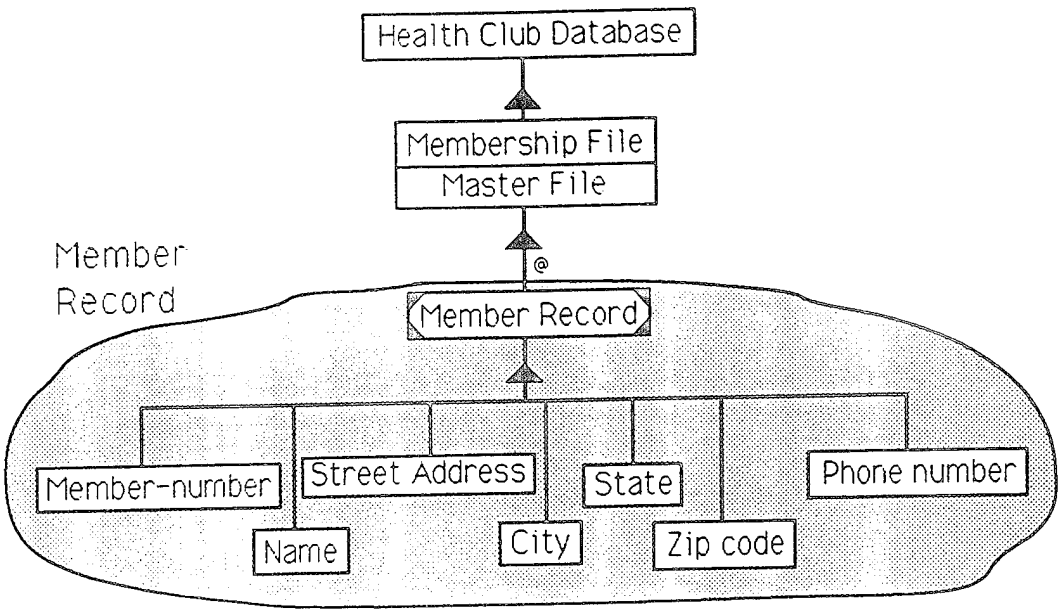


Fig. E-1 Health Club Example

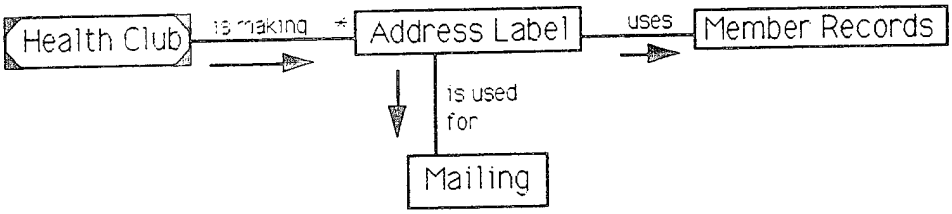


Fig. E-2 Mailing List

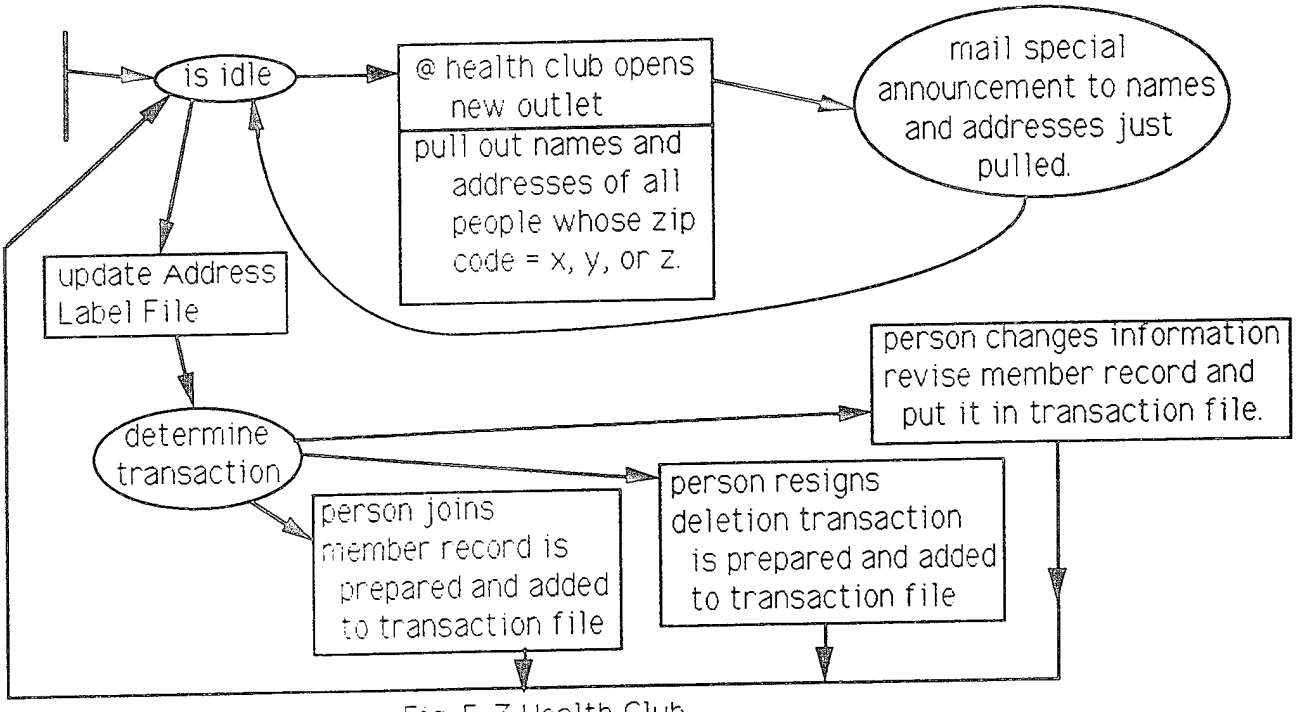


Fig. E-3 Health Club

Generated Text (Fig. E):

A Member Record is composed of a Member-Number, a Name, a Street Address, a City, a State, a Zip-Code, and a Phone Number. All Member Records compose a Membership File, or Master File. A Membership File composes a Health Club Database.

A Health Club is making Address Labels. An Address Label is used for Mailing. It uses Member Records.

The Health Club is idle. When the Health Club opens a new outlet, it pulls out names and addresses of all people whose zip code = x, y, or z. It then will mail special announcements to names and addresses just pulled. When the Health Club updates the Address Label File, it determines the kind of transaction. If a person joins, a member record is prepared and added to the transaction file. It then is idle. If a person resigns, a deletion transaction is prepared and added to transaction file. It then is idle. If a person changes information, then revise the member record and put it in the transaction file. It then is idle.

Appendix D

Generated Text

Data is processed by Computer Systems. By this we mean Data is batch processed by (n transactions at a time) and/or is transaction processed by (1 transaction at a time) by Computer Systems. Data is represented by Characters. A Letter, a Number, and a Special Character are all types of Characters. Examples of a Special Character are $, ?, and *. One or More Characters compose a Field. One of more fields compose a record. One or more records compose a file. One or more files compose a database. Data is organized into fields, records, files, and databases. Applications use Data from Databases.

Batch Processing is a Processing Technique. Batch Processing has Advantages. Examples of Advantages are inexpensive and efficient. Batch Processing has Disadvantages. An example of a Disadvantage is no quick response. Batch Processing uses a Master File and a Transaction File. Transactions compose a Transaction File. Addition, Deletion, and Revision are types of Transactions.

In the case of a Batch Processor, it waits for a transaction. If the number of transactions is greater than the number needed for an update, then update the master file with the transaction file. When a new transaction is needed, create a new transaction. If the transaction is addition, then create a new record and add the record and transaction to the transaction file. If the transaction is deletion, add the record to be deleted and the transaction to the transaction file. If the transaction is revision, them recall the record to be revised. Then revise the record and add the revised record and the transaction to the transaction file.

Transaction Processing is a Real-Time Processing Technique. It can affect the Activity at Hand. A Real Time Processing Technique must be On-Line. By this we mean, a Terminal is directly connected to a Computer. Transaction Processing has Advantages. An example of an Advantage is immediacy. Transaction Processing has become necessary for Service Industry. Magnetic Disk Development caused Improvements in Transaction Processing. Transaction Processing allows a User and a Computer to communicate directly.

A Transaction Processor will wait for a transaction need. When a new transaction is needed, decide the type of transaction. If the transaction is an addition, then create a new record and add it to the master file. If the transaction is a deletion, locate the record to be deleted and remove it from the master file. If the transaction is a revision, recall the record to be revised. Then revise the record and add the revised version to the master file.

A Magnetic Tape Unit reads and writes Data. Magnetic Tape Unit is composed of a Read/Write Head and an Erase Head. Both have an electromagnet. An Electromagnet manipulates Data. By this we mean, an Electromagnet reads, writes, and erases Data. Data is stored sequentially on Magnetic Tape. Magnetic Tape looks like Home Tape Recorder Tape. Both are made of Mylar Tape. Magnetic Tape has Dimensions. An example of Dimensions is 1/2" on 10 1/2" reel. Magnetic Tape has an Iron-Oxide Coating. An Iron-Oxide Coating has Extremely Small Magnetic Spots. Data is stored as Extremely Small Magnetic Spots. Magnetic Tape is a Storage Media. Magnetic Tape accesses Data. By this we mean, it waits for a need to access a record. When a need for a record is present, read the first record. If the record read is the record needed then it is finished. If the record read is not the record needed then read the next record. Data is read by a Tape Unit and put into Computer Main Storage. A Magnetic Tape Unit will wait for instructions. If the instructions are to read, then the electromagnet converts magnetized areas into electrical impulses. Then it sends the electric impulses to the

processor. Then the Magnetic Tape Unit returns to wait for instructions. If the instructions are to write, then the erase head erases the old data. The Magnetic Tape Unit will then wait for the processor to send the electrical impulses. At reception, the electromagnet converts the electrical impulses into magnetized areas. The Magnetic Tape Unit will then wait for instructions.

A Magnetic Disk is a Direct-Access Storage Media (DASM). A DASM sequentially and randomly stores Records. DASM is a Storage Media. Storage Media stores Data. Hard Disks and Diskettes are Magnetic Disks. A Hard Disk looks like a Large Brown Compact Disk. It records on a Top Surface and a Bottom Surface. A Hard Disk has a Diameter. Examples of Diameter are 14 inch, 5.25 inch and 3.5 inch. Three or more Hard Disks are organized into a Disk Pack. A Disk Pack looks like a stack of Stereo Records. It has space between the disks. A Disk Pack is a Type. A Type has N Platters. A Diskette has two Surfaces. A Surface has N tracks. A Track is a Concentric Closed Circle. Data is stored on a Track. A Surface has a Magnetic Oxide Coating. A Diskette is a Round Piece of Plastic. It has N Types. The same amount of time is needed to read the data on the outer track as on the inner track. The Type determines the Tracks. Hard Disks and Diskettes are used in a PC. A PC reads Data within N seconds. A Diskette directly accesses Data. The Direct Access increases Interactive Computing.

Appendix E

Algorithm for Natural Language Generation

This is the pseudo-code for the natural language generator in which I am designing. This code will later be translated into an OB model-design from which I will implement the design.

1. Wait for diagram.
2. Determine if diagram is OB model or OR model.
3. If OR model then:
   a) Identify Root
   b) Owner = Root
   c) Analyze untraversed relationships connected to the owner and create option list.
   d) Pick a relationship randomly from option list.
   e) Find any roles or participation constraints or time constraints attached to owner or recipient(s).
   f) Case relationship =
      1] Specialization - Generalization
      2] Binary
      3] n-ary
   g) If a recipient has instances, then describe them.
   h) If a relationship is a high level relationship, then move to low level diagram. Goto a).
   i) If a recipient is high level object, then move to low level diagram. Goto a)
   j) If recipient has untraversed relationships connected to it then owner = recipient. Goto c).
   k) If another recipient has any untraversed relationships connected to it then owner = that recipient. Goto c).
   l) If owner has untraversed relationships connected to it then Goto c).
   m) State notes and general constraints.
   n) Goto 1.
4. If OB diagram:
   a) Start at Initial Transaction
   b) State Initial State
   c) Analyze out arrows
      1] double - ended arrows have priority.
   d) Traverse the diagram using a pre-order tree traversal pattern. A branch ends when it runs into either a final state/transition or an already traversed item.

Appendix F

Rules of Generation

Overall:

These are the rules that I have established to generate the text following each diagram. These rules will be incorporated in the design of the final program.

1. Describe OR model before OB model unless otherwise stated.

2. Describe the low-level diagram of an object immediately after encountering that object for the first time.
   - If the low-level diagram is large (i.e. will take more than 5 sentences to describe) then describe the diagram in a separate paragraph.

3. The system of diagrams will have a translation table for different symbols. Default table would be:

Replace	With
=	equals
>	is greater than
<	is less than

etc.

4. If the subject of one sentence was also the subject of the previous sentence, use a pronoun (it, he, she) in place of the object's name.

5. If no priority has been established, randomly pick one of the eligible options.

6. Describe notes after all objects referenced before the note has been introduced.

7. If more than one diagram that is on the same level and in the same diagram group, then start a new paragraph when beginning on each subsequent diagram of the level.

OR Models:

10. Root object will be noted by a box with blackened corners. The root object will be the subject of the first sentence.

11. The PC is stated immediately before the name of the object within the sentence. If the PC is:

    * -> make object plural

n -> state number
@ -> all instances of the object participate in the relationship
n:* -> n or more
0:1 -> may or may not
‘ ‘ -> assume is 1; object remains singular

12. If an owner has more than 1 binary relationship with the same recipient, then describe all of them in one sentence.

13. If an object has a role name, then that role name is stated between the PC and the object name.

14. If an object has an instance of itself, describe it immediately after the object is first encountered (before description of low-level diagram of that object).

An example of (object name) is (example.
Examples of (object name) are (example) and (example).

15. If a relationship has a time constraint, then describe constraint either at the beginning or at the end of the sentence.

Within (time constraint), (object) (relationship) (object).
(object) (relationship) (object) with (time constraint).

16. If there is a low-level diagram of a relationship, describe the low-level diagram in the next sentence.

By this we mean, (description).

17. If an object doesn't have a relationship emanating from it:

(a) back track through objects until an object with an unencountered relationship emanating from it is encountered.

(b) describe in reverse a relationship that is connected to it.

18. If a binary relationship has many recipients:

(a) describe after the subject is encountered.

(b) describe after all recipients are encountered.

19. Describe relationships involving exploded object before describing relationships involving objects within the exploded object.

20. General Constraints should be stated after all items involved in the constraint have been introduced.

Priority/Rule for OR Models

(Order in which relationships should be described.)

21. Generalization - Specialization

(Specialization class name) is a (Generalization class name)

22. Binary

(Object class name) (relationship) (object class name)

23. N-ary

State the N-ary relationship as it is written on the diagram.

24. Aggregation

(Aggregate class name) is composed of (Sub-Part class name)

* (Sub-Part class name) composes (Aggregate class name).

25. Association

(Member class name) is a member of (Set class name).

* (Set class name) has (Member class name) as a member.

* Reverse description of the relationship. Forward description has priority though.

OB Models:

26. To describe an OB model with more than two sentences, chronological transition words should be randomly used.

27. Transition may be described by either trigger then action or action then trigger. If are than 2 actions, the describe the trigger first then the actions.

If trigger - action, replace @ with when.

Message-Oriented Middleware (MOM): A Key Technology for the Successful Deployment of Distributed Client/Server Information Systems

Gail V. Quigley

AT&T Global Information Solutions

1. INTRODUCTION

IT professionals tasked with the development, deployment and support of the emerging DoD information systems infrastructure have the following hopes and dreams:

- To integrate all current and emerging information systems into a standards-based, open systems architecture employing modern databases, client/server, and graphical user interface technology.
- To provide users of the information system with access to a single logical repository without regard to the technical characteristics and location of the system or systems in which the information resides.
- To enable incremental improvement, re-deployment, and replacement of underlying information systems while preserving the user view of information as a single logical repository.

The placement of data and processing on geographically dispersed computers presents new challenges for systems development, deployment and support. For example:

- The availability of distributed processing nodes is unpredictable.
- Nodes on the network could be a mobile workstation.
- The bandwidth of the network impacts the placement of processes and data.
- New processes and databases must be deployed with little or no impact on the operational system.
- A single view of the entire set of distributed system components is required to reduce the complexity of systems administration.

Middleware products are emerging to address these challenges. Middleware is important in the integration of heterogeneous computing environments. It enables developers to

build software components without being concerned with the underlying complexity of database servers, network interfaces, and procedural nuances, etc. This paper describes:

- A class of middleware called Message-Oriented Middleware (MOM).
- TP Monitor usage as a MOM solution.
- Two-Tier vs. Three-Tier Distributed Client/Server Architectures.
- The Features and Functionality of the TOP END TP Monitor.
- Two large enterprises that are using the technology.

2. MESSAGE-ORIENTED MIDDLEWARE (MOM)

The term *middleware* has become a catchword for a software layer between applications logic and distributed computing services. MOM is a type of middleware that is based on a message passing model. Since Transaction Processing Monitors (TP Monitors) include a message passing model based on the notion of a transaction, they can be easily adapted to the MOM model. Middleware, MOM, and the use of TP Monitors to implement a MOM are discussed in this section.

A Middleware Definition

As shown in the figure below, middleware is software layered between your application logic and the underlying networking, security, and distributed computing technology. It provides all of the critical services needed to manage the execution of applications in a distributed client/server environment. Middleware shields your applications from much of the complexity of distributed computing and can reduce the cost of moving from the closed, glasshouse computing of the past to today's more common distributed, open computing environment.

Middleware is the key piece of technology required to integrate applications running on diverse platforms across enterprise networks. While many users are showing an interest in the Open Software Foundation's (OSF) Distributed Computing Environment (DCE) as a distributed computing platform of choice, users who are really building distributed client/server systems are deploying defacto middleware products.

Authentic client/server systems are built as extensible, scaleable, and easily maintainable applications that operate across hardware and software platforms , and leverage the inherent strengths of the client and the server. Users still operating in the mainframe mode use DCE 's Remote Procedure Call (RPC) to pass blocks of data back and forth between client and server. A more appropriate interface between client and server is a functional interface that supports the loosely coupled process model that is the trademark of true

client/server systems. The interface must deal with layers of client and server application software, many different flavors and versions of operating systems software, networking variations, and variable hardware and software configurations.

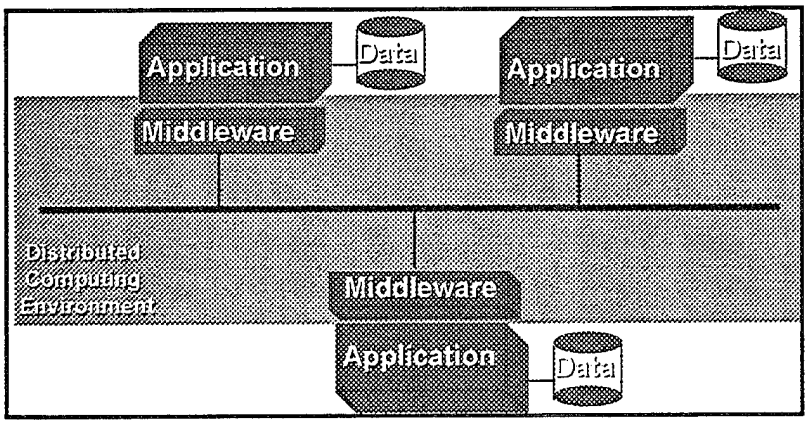


Figure 1: Middleware - the Foundation Layer Tying Together All Enterprise Components

Defacto middleware falls roughly into three categories: object request brokers, message-oriented middleware (MOM), and Unix-based transaction managers. Of the three, transaction managers have the greatest chance of dominating the middleware market, judging by the current number of customers and maturity of the product lines.

As with most technologies, the services that middleware performs are largely vendor specific. Product focus includes:

- transaction monitoring,
- dynamic load balancing,
- legacy system access,
- enterprise-wide administration,
- and end-to-end security.

MOM and the Transaction Processing (TP) Monitor

MOM is middleware that allows independent applications to exchange data simply by sending messages. The MOM system ensures that messages are delivered by using reliable queues and by providing directory, security and administrative services required to support messaging. The MOM technology provides a single uniform environment for building and maintaining distributed applications across diverse computer platforms, database servers, and networks.

TP Monitors are the most mature, and are beginning to dominate the middleware market. Oddly, the user community has discovered that transaction managers have more value as a

client/server interface capability than as transaction processing servers. Most use it to ease the process of developing, managing, and connecting distributed heterogeneous applications. Consequently, most vendors of TP Monitors have recently reintroduced their products as "client/server message-oriented middleware."

TP Monitors have been a required component of mainframe-dependent MIS departments. Because organizations want to move some of their on-line transaction processing applications to open, distributed environments, TP Monitors have been introduced in the open systems domain.

Also some of this migration can be ascribed to database vendors who have been developing low-end tools that can serve as transaction processing surrogates. These database vendor tools do not scale well over large enterprises. "The DBMS vs. TP Monitor Debate," a report published by the Stamford, Conn.-based Gartner Group asserts that TP monitor middleware is unnecessary if an organization's network has less than 50 users, has databases with less than 2Gb of data, and processes fewer than 25,000 transactions daily. Organizations with small networks can get OLTP-like functionality from any number of database management system toolsets. However, once your enterprise grows beyond this size, a middleware solution such as a message-oriented TP Monitor should be considered.

3. TWO-TIER VS. THREE-TIER DISTRIBUTED CLIENT/SERVER ARCHITECTURES.

The real issue is whether all business application logic throughout the enterprise is database centric or distributed. With a database solution users develop applications using SQL (typically with non-open extensions) or vendor-defined (i.e. proprietary) stored procedures which are coded directly into client application logic. Doing so has a number of ramifications:

- From a performance point of view executing SQL directly from the client may limit performance, because of the connection-oriented communication behavior (lots of communication and database resources are required for each user) and because typical SQL commands require more network traffic.
- With SQL or stored procedures in the client you lose the isolation between client and server application logic.

- SQL is not SQL is not SQL, and similarly for stored procedures. Thus client logic is simply not portable to different vendors databases. Nor are such constructs typically portable through vendor's "open" connectivity products, like database gateways. And few large users are willing to lock in all their business logic to one vendor's proprietary constructs because this limits their application development choices to those offered by a database vendor.
- Although it is convenient from an administration point of view, having all your business logic down in the database increases the workload on what is undeniably the largest user of system resources and typically the hottest spot in your enterprise: the database engine! It is more desirable to push business logic out of the database -- not force more down into it!

In the Three-Tier model your business logic may be developed using an unlimited variety of open programming languages and development tools, and the logic may be executed separately from the database, thus better distributing your distributed application workload. In the Three-Tier model you may develop your business application logic in a portable fashion, then develop separate logic exclusively for database access using any portable or proprietary means you desire. This increases application performance and overall design flexibility.

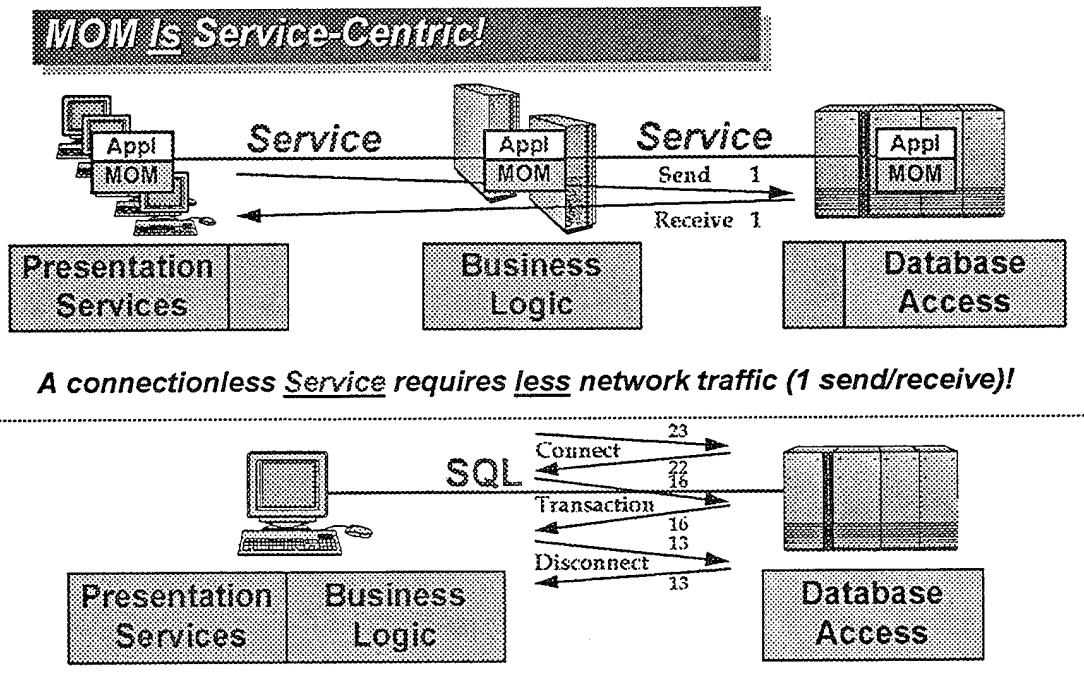


Figure 2: Three-Tiered Service Centric vs. Two-Tiered Database Centric Architecture

4. FEATURES AND FUNCTIONALITY OF THE TOP END TP MONITOR

TOP END functions are divided into several modular components that are designed for a distributed, message-passing environment. In TOP END terminology, a "component" is a process or logical group of processes that performs a function. TOP END components are broken down into *on-line* components that work together to process distributed transactions during the on-line day, and *tools* that are essential to define and manage an open distributed environment.

On-line Components

Application Components.

The fundamental TOP END component is the *application component*. Application components are used to create and grow distributed TOP END systems. Applications reside in the application components, and it is the application components that are distributed on the various nodes throughout the network. The applications could be user-written, provided as a solution by an independent vendor, or supplied by AT&T as part of the TOP END base product. A comprehensive set of services and libraries are available to an application component. An application component can begin and end distributed transactions, interface with resource managers, interface with communication managers, and more.

Node Manager.

The second key on-line components is the *node manager*. The node manager is a collection of processes that offer core services to allow TOP END processing to take place on a given node. These processes, in general, work independently of each other. Processes on a TOP END node that share node manager services include application components, network interface processes, and logging and transaction services. Services provided by the node manager include transaction management (for example, commit coordination), logging, failure recovery, client/server request handling, security management, runtime administration, and application component control.

Network Agents.

Network agents are used to allow transactions and service requests to enter a TOP END system from an application or networked workstation that does not have a node manager on it. TOP END's Remote Client support uses a network agent.

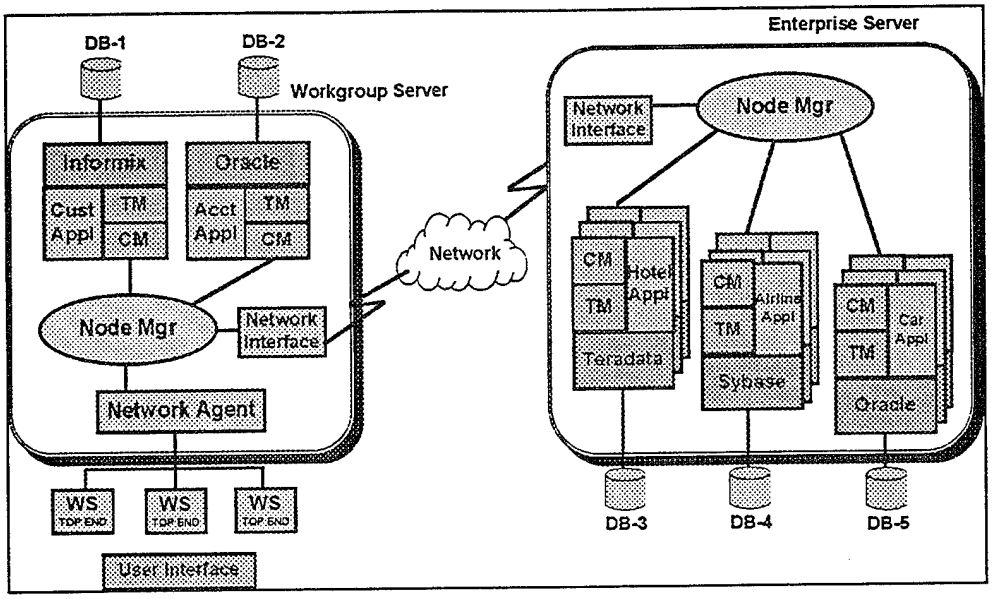


Figure 3: Top End Components

Login Clients.

The *login client* is a special case of a TOP END application component that uses a library of terminal support routines and TOP END screen formats to facilitate communications between a character-mode terminal (client) and a TOP END application program (service). The login client is a format driver that that gives terminal users a highly interactive interface with which to access the available application services. Login clients perform screen management and mapping with field level validation.

Personal Computer Support.

PCS, or personal computer support, provides a highly interactive interface that is similar to, but richer than, the login client. PCS operates on a DOS personal computer. (Note that TOP END also offers DOS workstation-based application support.)

Tools

In addition to providing the components and services needed to execute applications in an open, distributed environment, TOP END provides a myriad of tools for managing the environment.

Administration.

TOP END provides a full range of graphical, menu based administrative tools that make it easy to manage distributed runtime environments. The runtime administration tools are used to perform component start-up and shutdown, manage auditing and recovery, activate communication links, perform automatic software distribution, and so on. For high availability reasons, there are two levels of administration provided: global and local.

Global administration provides a single operational view of the enterprise allowing administrators to control all system nodes from a single workstation. For certain failure situations, however, *local* administration can be used to accomplish controlled administration on a node-by-node basis.

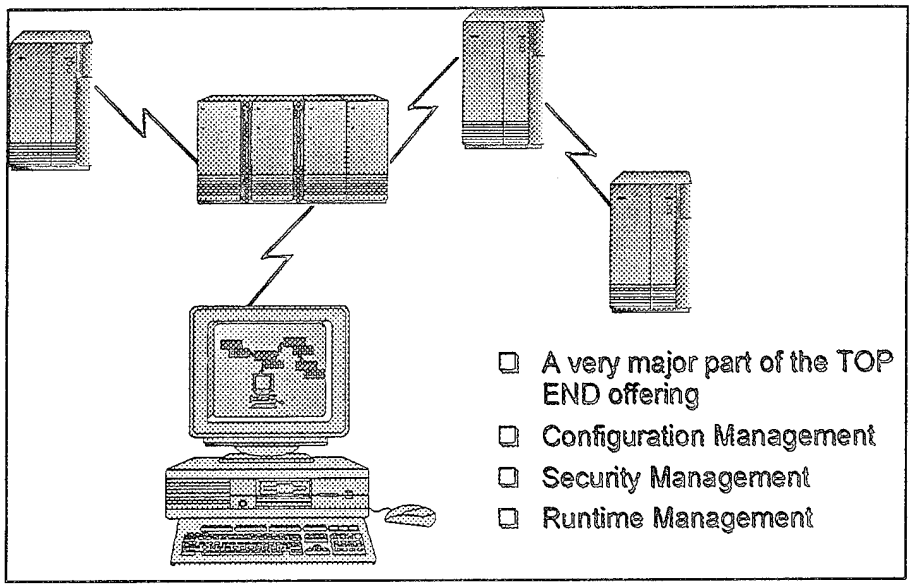


Figure 4: Control All Nodes from a Single Workstation

Interactive System Definition.

The *Interactive System Definition* (ISD) tool provides a user-friendly, Windows interface for defining TOP END systems. It is composed of two parts: system definition and system generation. System definition is used to define and specify the components that the administrator would like to include in a TOP END system. System generation then extracts the needed definitions for the desired TOP END system so they can be placed on the host. Once on the host, system generation automatically performs the linking needed to create the component executables.

Format Management.

TOP END *format management* tools allow screen layouts to be easily created and customized for a wide variety of devices. One of the format management tools is an interactive screen builder. The screen builder allows an application developer or administrator to define custom screens that can be used with their application components. The screens are compiled and stored in format libraries. There are also tools for creating, copying, and reporting format libraries.

Application Development.

TOP END lets you choose the development environment that works best for you. It supports everything from Native C, C++ and COBOL to high level 4GL client/server

development tools. The *TOP END Development Environment* offers programmers a variety of integrated graphical tools for building applications in a building block approach. Using this environment, you can create application components and include them in multiple systems without having to reenter the data.

Key Features of TOP END

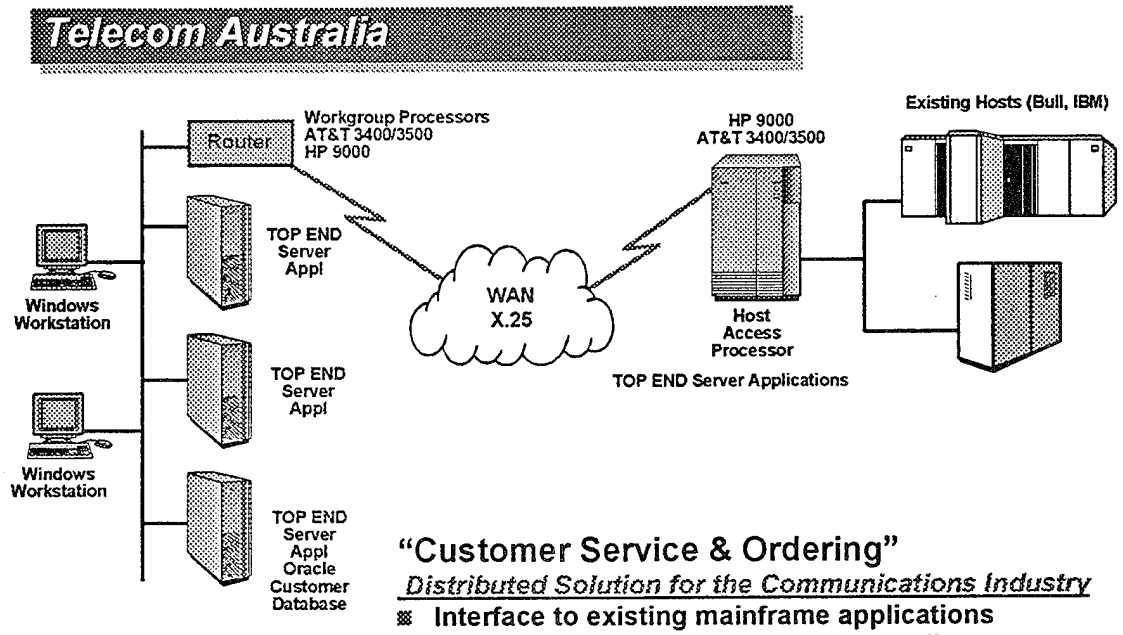
TOP END middleware is considered "robust" because it provides an environment rich in features like the following:

- **Distributed Transaction Management.** TOP END's method of distributed transaction management is based on the X/Open DTP model.
- **Support for Client/Server Interaction**. TOP END offers services for cooperative processing that allows clients and servers to participate in a distributed transaction. Either the client or server can begin a transaction. The clients and servers can be on the same or different nodes, and the location of the server is transparent to the client. TOP END offers advanced security capabilities to control the client's access to servers.
- **Dynamic Workload Balancing.** TOP END automatically generates and manages parallel copies (replicas) of applications and performs all the needed workload balancing among the copies to ensure that they are all evenly utilized.
- **Recoverable Transaction Queuing.** To enhance the ability for distributed applications to work together in a "connectionless" fashion, TOP END performs recoverable transaction queuing (RTQ). RTQ is a modular store-and-forward capability that enables complex multi-step computations to be broken down into a series of transaction steps, linked via durable transaction queues. RTQ prioritizes queues based on message context, message content, time of day, and hooks for user defined request scheduling algorithms. RTQ guarantees delivery of messages, allowing distributed applications to work together in a "connectionless" fashion.
- **Application Parallelization.** By automatically replicating and distributing application components throughout your enterprise, TOP END "parallelizes" your applications for you, without any special programming efforts or additional resources.
- **Multistep Transactions.** Many transactions require multiple steps to complete. TOP END processes multistep transactions as a single transaction so that aborting a transaction rolls back the entire creation, not just the last step.
- **Two-phase Commit Processing.** TOP END supports transactions that have steps processed on different processing nodes in the network by using a two-phase commit process that is completely transparent to applications.

- **Message-Sensitive Routing.** TOP END routes messages based on their context. This allows transactions to be processed where the data is, rather than sending the data to the transactions.
- **Automatic Software Distribution.** TOP END runtime administration automatically distributes software from the administrative node to other TOP END nodes.
- **Automatic Recovery.** The TOP END system is designed to be highly available and require as little operator intervention as possible. It provides automatic recovery from application failures, transaction failures, network failures, and node failures.
- **Mainframe-class Safety Nets.** To secure your move from legacy computing to open systems, TOP END provides the same type of safety nets as traditional glasshouse environments. TOP END provides database consistency, transaction recovery, fault tolerance, application resilience, and enhanced security.
- **Multiple Database Support.** TOP END is specifically designed to work with third-party database management systems (DBMSs) without imposing its logging, locking, or commit protocols. It supports application database processing by automatically opening, closing and recovering schemas so that application programs do not have to. TOP END supports leading DBMS systems including Oracle, Informix, Teradata, and Sybase. When more than one database is used, TOP END will update each database with a single transaction.
- **Network and Location Transparency.** TOP END tracks the location of all components in the distributed system so that they remain equally available to users. That is, users have access to all resources and applications in the TOP END system, regardless of where they reside.
- **Scalability.** Because of TOP END's modular design, you can add new applications, new users, and new processing nodes to your enterprise system any time you want. You can add these components one at a time or in clusters, and you can do so without disrupting current operation. Since TOP END operates in a distributed computing environment, you can have as many workstations and servers as you need to fulfill your current business demands.
- **Open Systems Compliance.** To allow users maximum interoperability and portability with related products from other vendors, TOP END uses standard open system protocols. The use of a standard interface allows applications to be independent from TOP END, thus fostering fault isolation and recovery. In addition to supporting all protocols established by X/Open Limited, TOP END complies with OSI- and IEEE-based standards.

5. SAMPLE USAGE IN A LARGE, DISTRIBUTED ENTERPRISE

Telecom Australia

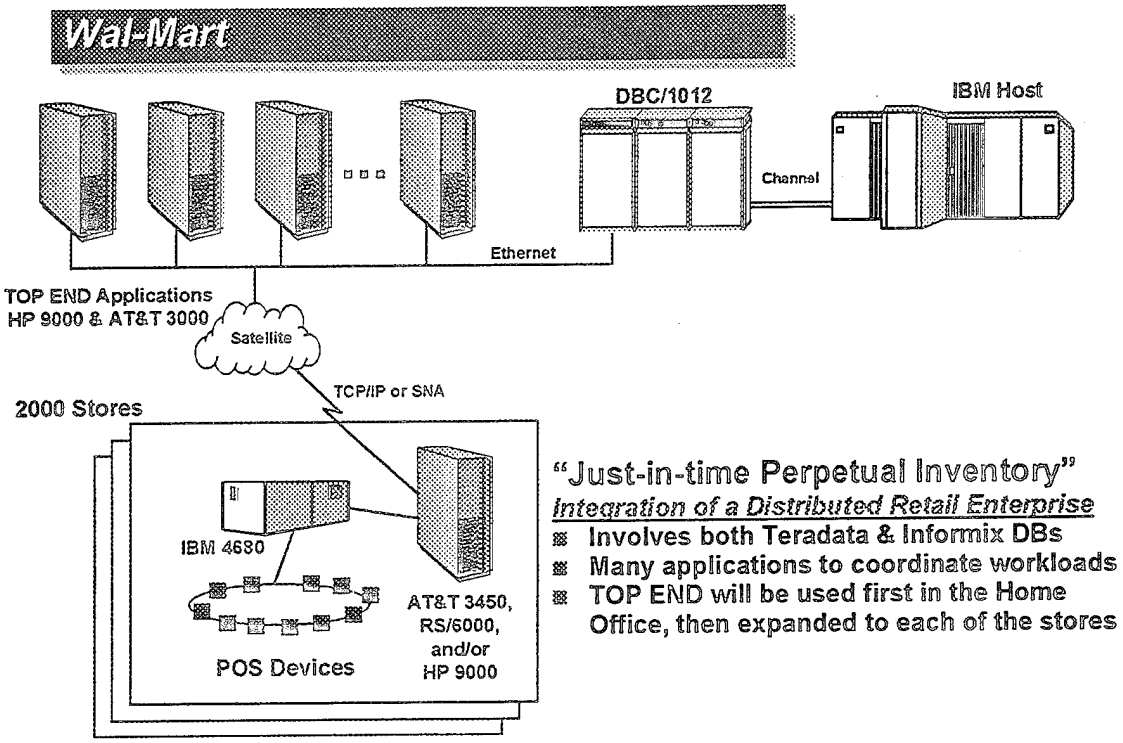


"Customer Service & Ordering"

Distributed Solution for the Communications Industry

- Interface to existing mainframe applications
- TOP END Microsoft Windows remote clients
- Cooperative processing between hosts and Branches
- Utilize workgroup servers to validate customer order requests

Wal-Mart



6. CONCLUSIONS

For years, large enterprises have struggled to make their disparate computing components work together in a fashion that is meaningful to them. Often this feat requires connectivity of heterogeneous databases, multiple communications and networks, different hardware platforms, and more. More often than not, seamless connectivity was simply not possible for enterprises such as these. Now, with the movement towards open systems, there are middleware products available that can integrate the enterprise computing components.

7. BIOGRAPHIC SKETCH OF THE AUTHOR

Gail Quigley has been a computer professional for 30 years. After completing a Bachelor of Arts degree in Mathematics, she began her career as a systems programmer at the National Security Agency. She then joined Control Data Corporation where she held various technical and management positions during her 17 years of service at the Federal Systems Division, International Operations and Corporate Headquarters. Her major areas of expertise were data communications and data management. She was an independent

consultant for 4 years. Her clients included the Department of Energy, Combustion Engineering, NIST, IDA, FNMA, etc. She joined Teradata Corporation which was acquired by NCR and then AT&T. She was a Product Marketing Manager for AT&T GIS in the area of Network and Enterprise Computing. Her present position with AT&T GIS is as a Senior Consultant for the DoD Team of AT&T GIS Federal and State Team.

Gail Quigley, AT&T Global Information Solutions, 2 Choke Cherry Rd., Rockville, MD 20850, Phone: 301-212-5102, Fax: 301-212-5151, E-mail: Gail.Quigley@WashingtonDC.ATTGIS.COM

Phillips Laboratory's Technology Transfer Database

Andrea E. Gleicher
USAF Phillips Laboratory
and
Lila A. Hicks
The Aerospace Corporation

1. INTRODUCTION

The Technology Transfer Database program was developed by the Air Force Phillips Laboratory under the direction of the International and Industrial Programs Division. The Aerospace Corporation provided the initial design and implementation of the program, with software development support from Tech Reps, Inc. in Albuquerque, New Mexico.

The program is a tool for organizing and showcasing descriptions of technology and technology transfer activities within the Lab.

2. BACKGROUND

In the past, the Air Force laboratories have concentrated their efforts on developing technology to meet national defense needs. This is still our primary mission, although it is clear that much of the technology developed by the labs can also be used in commercial products and processes to make U.S. industry more competitive in the global marketplace. This realization has become an important idea for the Phillips Lab and other Air Force research and development organizations. We are now working to transfer these technologies outside the Department of Defense. In this new environment, the Air Force is working to find new partners and clients and to be responsive to the demands of U.S. industry. Toward this end, we need to let people know who we are, what we do, and how we may be able to help them.

As the importance of this new technology transfer mission grew for the Laboratory, it became evident that a new tool was needed (see Figure 1):

- To convey to outside organizations the types of research performed at the Phillips Laboratory including the Laboratory's technology transfer activities.
- To highlight successful programs, unique research and test facilities, specialized expertise and centers of excellence within the Lab, and successful technology transfer efforts, such as commercialization of a new technology.
- As a management tool to provide descriptive information and metrics about laboratory efforts for responding to requests from higher headquarters.

- As an internal tool for tracking the history of activity of a cooperative effort from initial contact through establishment of a formal relationship (e.g. cooperative research and development agreement, memorandum of agreement, educational partnership, etc.) through accomplishment of the effort.

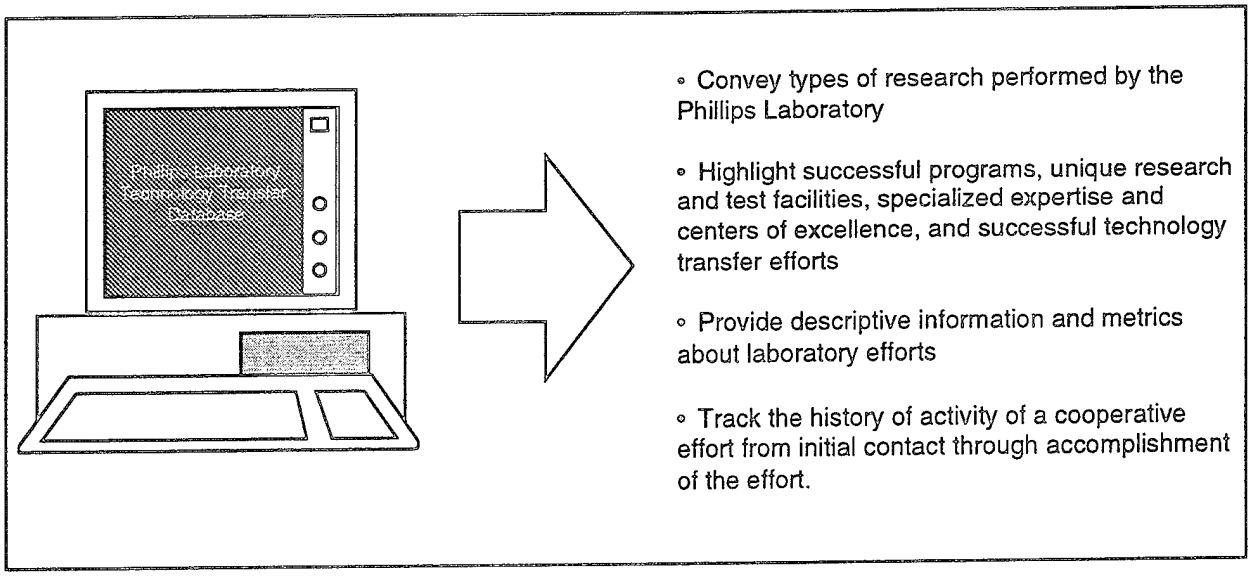


Figure 1

3. DEVELOPMENT HISTORY

To answer these needs, a prototype of the database was developed based on a similar effort done by the Department of Energy's Ames Laboratory. The prototype was developed using commercial off-the-shelf software, in this case the 4th Dimension database application, and hosted on the Macintosh platform. Initial design and population of the database took about nine months. The database was shown at various technology transfer conferences and was briefed to other Air Force laboratory technology transfer representatives as well as Headquarters, Air Force Materiel Command (HQ AFMC). At the request of these organizations, the database was reworked for the PC platform using FoxPro for Windows. In March of this year, Version 1.0 was completed and a runtime shell was delivered to each of the Air Force laboratories to evaluate its usefulness in their respective organizations. Since March, final corrections and improvements have been completed, as well as the addition of a contact tracking module.

4. DATABASE DESCRIPTION

There are two modes of operation for viewing and working with the database.

- View-only, or conference mode, is intended for use by outside individuals. This mode provides a user-friendly environment which allows the user to navigate through the database by way of buttons and by clicking on items of interest dis-

played in a list format. Searches of the database for specific areas of interest and generation of standard reports are also permitted.

- Maintenance mode is for use in entering, modifying and deleting records and linking table information (for example, specifying a point of contact for a given program or technology transfer effort). This mode has a user-friendly, menu-driven interface which also allows import of data from other files and applications.

(Note that the runtime shell which was delivered to other Air Force laboratories allows for complete functionality of the database to include entering, modifying and viewing data; searching records; and generating standard, predetermined reports. The runtime shell does not require that FoxPro be installed on the computer which is running the database. However, in order to make any changes in the database structure, generate specialized reports, or do complex queries and searches, FoxPro is required.)

The database information is displayed in five major areas: Programs, Facilities, Capabilities, Technology Transfer, and Directory (see Figure 2). The layout and functions of the Programs, Facilities, and Capabilities areas are similar in that they include the name, technical category or area, and a description of the item. Additional fields, such as location in the case of facilities, are included as applicable.

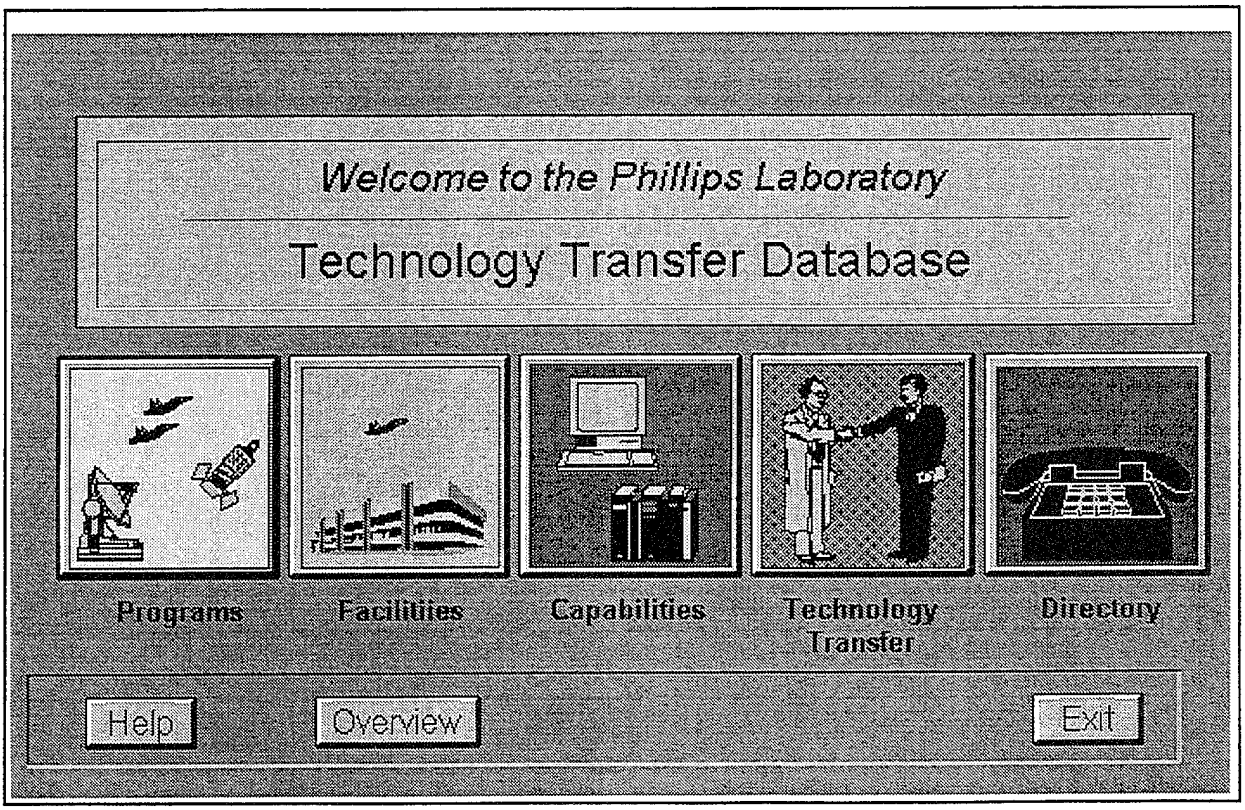


Figure 2

The Technology Transfer area is a little different in that it is further broken out into types of technology transfer efforts such as Cooperative Research and Development Agree-

ments (CRDAs), Small Business Innovative Research (SBIR) efforts, and Patents. Each of these areas are similar in how they are accessed and the information conveyed, but individual fields vary somewhat depending upon the type.

Records in the Programs, Facilities, and Capabilities areas may be related to records in the Technology Transfer area and the related information displayed. For example, if a given program has resulted in a patented piece of hardware or process, that patent description may be linked to the record for that program. As another example, a given CRDA effort may make use of a specific Laboratory facility; if so, these two records may be linked. While browsing the database, the user can view linked records so that a more complete picture of all related information is received.

The Directory area includes a listing of key Phillips Laboratory personnel by directorate, division and branch. This provides a way for the user to locate expertise for a given technical area of interest and narrow down his or her search from a broad area, such as Space and Missiles Technology, to a very specific topic, such as Precision Structures. The Directory includes name, title, organization, address, phone and fax numbers, and e-mail address fields.

The contact tracking module is a stand-alone feature to hold information about technology transfer and marketing activities (Figure 3). The Laboratory receives hundreds of calls for information each year as a result of conferences, exhibits, advertising and referrals. Contacts can be entered into the system to ensure prompt and efficient follow up. By generating a report once a week, it is possible to make sure all requested information has been sent to the appropriate individual or organization.

The tracking module includes a section for listing actions to be taken as well as those completed. As work with a contact continues, the database is updated so that a complete history is maintained. The technology transfer division of the Lab uses this as a way to follow progress on CRDAs, patents, MOUs and other transfer types. It also contains information on educational outreach programs and other types of cooperative efforts. This system provides the capability to produce reports on numbers and types of technology transfer efforts with the click of a mouse versus searching and compiling from paper files, thus reducing redundancy of work and time required to answer requests for metric type information.

A simplified version of the database table structure is depicted in Figure 4.

5. FUTURE OF THE DATABASE

The Phillips Laboratory database will soon be available in CD ROM format in addition to residing on the Internet via the Phillips Laboratory home page. We are also working to add multimedia capability to include sound and video as well as an overall introduction to the Laboratory.

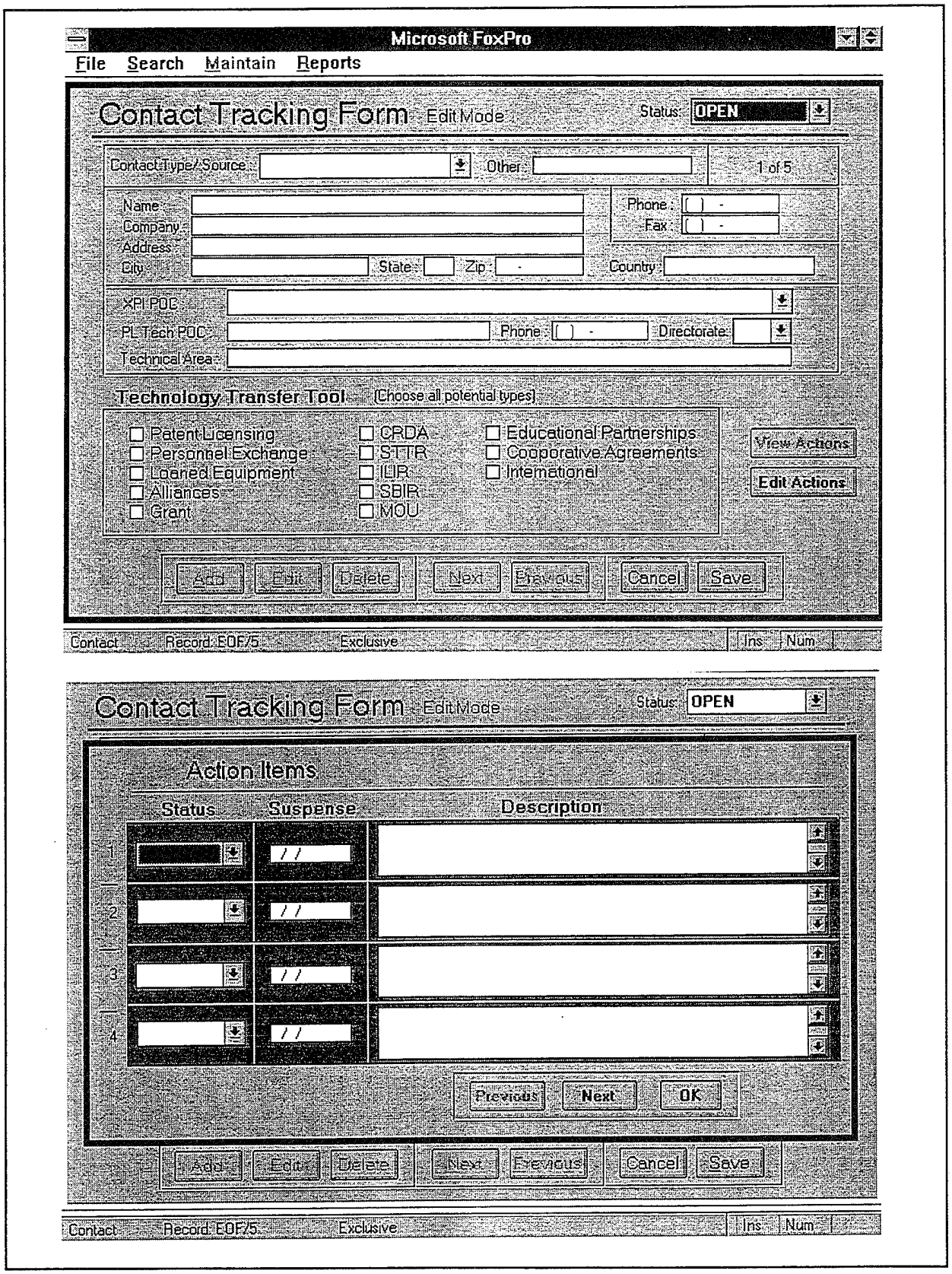

Microsoft FoxPro
File Search Maintain Reports
Contact Tracking Form Edit Mode
Status: OPEN
Contact Type/ Source:
Other:
1 of 5
Name:
Company:
Address:
City:
State:
Zip:
Phone:
Fax:
Country:
XPI POC:
PL Tech POC:
Phone:
Directorate:
Technical Area:
Technology Transfer Tool (Choose all potential types)
Patent Licensing
Personnel Exchange
Loaned Equipment
Alliances
Grant
CRDA
STTR
ILIR
SBIR
MOU
Educational Partnerships
Cooperative Agreements
International
View Actions
Edit Actions
Add
Edit
Delete
Next
Previous
Cancel
Save
Contact
Record: EOF/5
Exclusive
Ins
Num
Contact Tracking Form Edit Mode
Status: OPEN
Action Items
Status
Suspense
Description
1
2
3
4
Previous
Next
OK
Add
Edit
Delete
Next
Previous
Cancel
Save
Contact
Record: EOF/5
Exclusive
Ins
Num

Figure 3

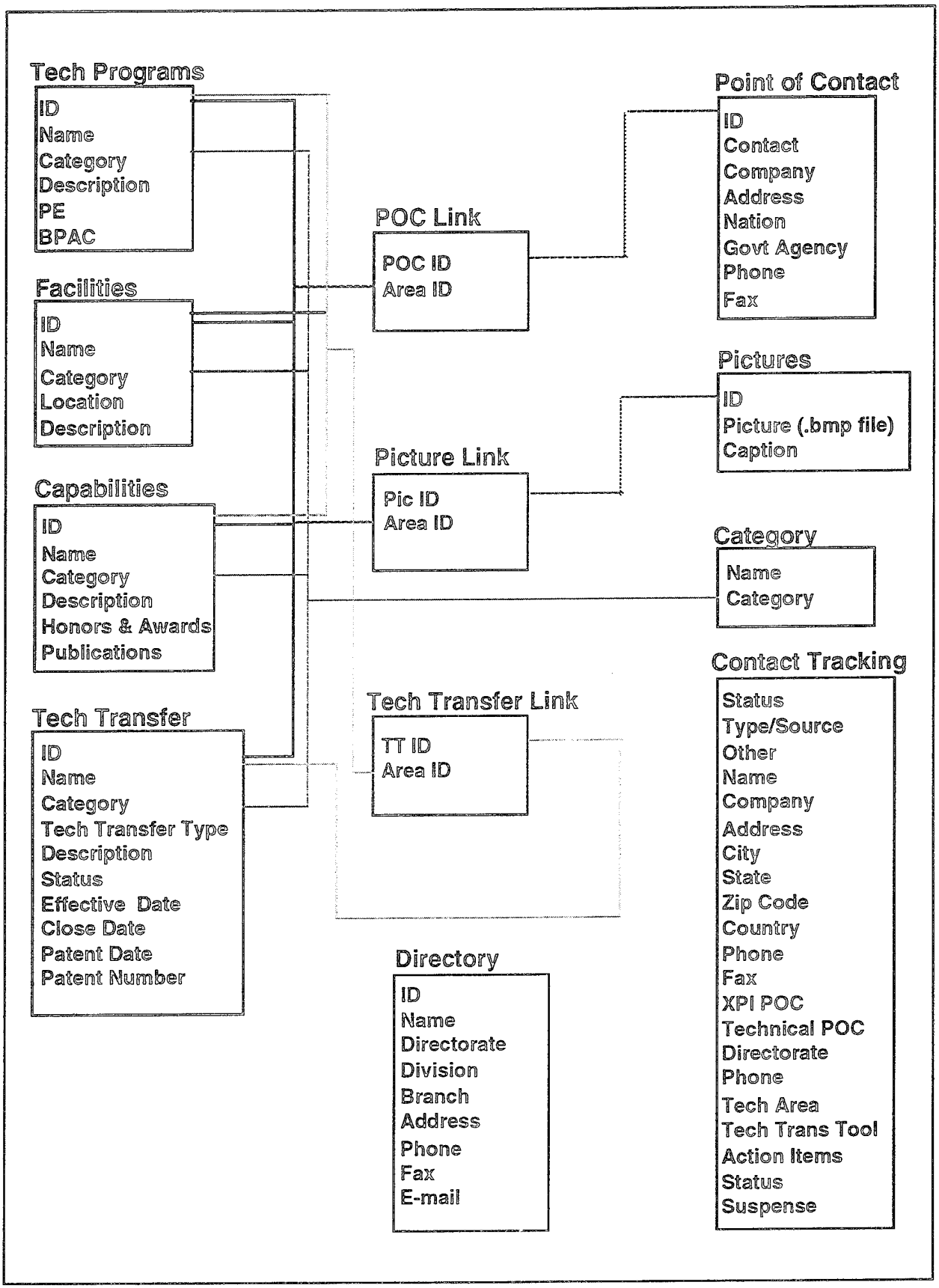

Tech Programs
ID
Name
Category
Description
PE
BPAC
Facilities
ID
Name
Category
Location
Description
Capabilities
ID
Name
Category
Description
Honors & Awards
Publications
Tech Transfer
ID
Name
Category
Tech Transfer Type
Description
Status
Effective Date
Close Date
Patent Date
Patent Number
POC Link
POC ID
Area ID
Picture Link
Pic ID
Area ID
Tech Transfer Link
TT ID
Area ID
Directory
ID
Name
Directorate
Division
Branch
Address
Phone
Fax
E-mail
Point of Contact
ID
Contact
Company
Address
Nation
Govt Agency
Phone
Fax
Pictures
ID
Picture (.bmp file)
Caption
Category
Name
Category
Contact Tracking
Status
Type/Source
Other
Name
Company
Address
City
State
Zip Code
Country
Phone
Fax
XPI POC
Technical POC
Directorate
Phone
Tech Area
Tech Trans Tool
Action Items
Status
Suspense

Figure 4

As mentioned earlier, the other Air Force laboratories — Wright, Rome, and Armstrong — are evaluating the utility of the database for their organizations. Once these separate database are populated, a link could be made from each of them to a common server. At this point, you would have a place that contains descriptive information on everything the Air Force superlabs are working on, and it would be accessible to millions of people around the world.

We also plan to coordinate with HQ AFMC to determine interest in expanding the database's use to other AFMC organizations such as the product divisions and logistical centers. If this were done, the logical evolution would be to link each organization's database on a common server under the direction of HQ AFMC (see Figure 5). This would allow a user to explore research and development, acquisition, test, maintenance and logistics information for the entire Air Force in one place allowing for leveraging of ideas and avoiding duplication of effort. It would also provide a tool for the laboratories, other DoD and government agencies, and industry to see the Air Force's programs and their interactions.

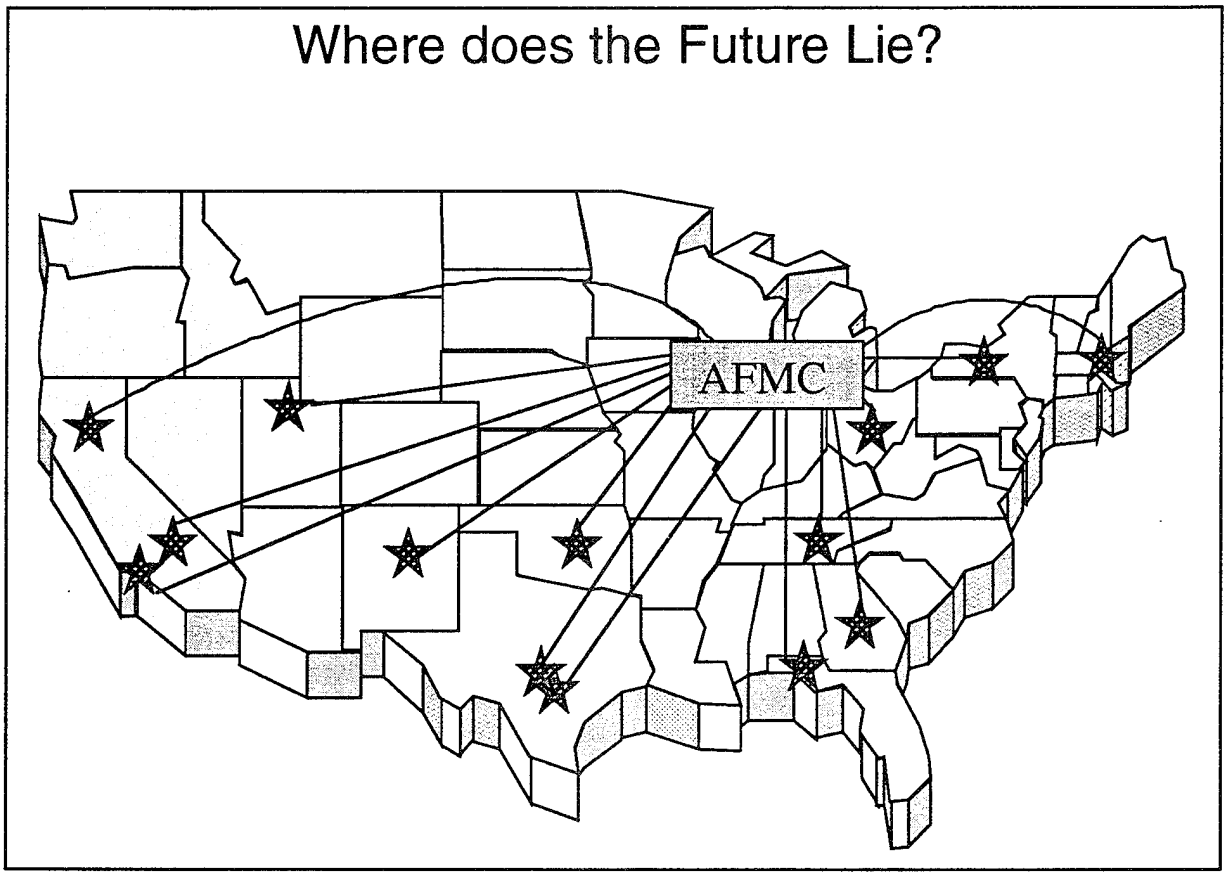


Figure 5

For more information about this program, contact:

Andrea Gleicher
Air Force Phillips Laboratory
PL/XPI
Kirtland AFB, NM 8711732
(505) 846-8932

Andrea E. Gleicher

Ms Andrea Gleicher works for the Industrial and International Programs Division of the Air Force Phillips Laboratory in support of the Technology Transfer Office. She assists in the overall marketing strategy to promote Phillips Laboratory research and development capabilities. In addition, she is responsible for econometric modeling to analyze the impact of government technologies on industry, and to determine the most effective mechanisms for the transfer and commercialization of technology. Prior to that, as a student, she was Chief of the Foreign Travel office while completing a Bachelor's degree in Economics at the University of New Mexico.

Lila A. Hicks

Ms Lila Hicks is a Senior Member of the Technical Staff of the Aerospace Corporation, Space Technology Directorate, at Kirtland AFB, NM where she provides technical support to the Air Force Phillips Laboratory's Office of Research and Technology Applications (ORTA). She began her career in the United States Air Force in 1981, assigned to the Deputy for Technology at Air Force Space Division Headquarters in Los Angeles, California. There she became involved in technology planning and development for future Air Force space systems. In 1983, Ms Hicks was transferred to Kirtland Air Force Base to help establish the Air Force Space Technology Center, precursor to the Phillips Laboratory. Prior to joining the Aerospace Corporation, Ms Hicks worked for SENTEL Corporation performing operational and joint service testing in support of the Air Force Air Warfare Center, the Air Force Operational Test and Evaluation Center and the Office of the Secretary of Defense. She holds a Bachelor of Science degree in Industrial Management from the Georgia Institute of Technology and a Master of Business Administration from the University of New Mexico.

Database Replication and Synchronization in the Global Command and Control System

Ron Harris, SRA Technical Services Corporation

1. OVERVIEW

1.1 Introduction

This technical paper describes the analysis and prototyping of various products to satisfy the Joint Operation Planning and Execution System (JOPES) data distribution/replication requirements within the Global Command and Control System (GCCS). The initial analysis focused on two products: Oracle's Symmetric Replication and Sybase's Replication Server, specifically their Oracle Log Transfer Manager (OLTM). SRA worked closely with both Oracle and Sybase engineers in prototyping and demonstrating their replication products (beta-releases) for government review at SRA's Fairfax, VA facilities.

This paper summarizes the extensive analysis documented in SRA's JOPES Database Distribution (DBD) Alternatives Study, May 15, 1995.

1.2 Background

To understand GCCS JOPES's data distribution and replication requirements better, some background information concerning the current JOPES system is provided below.

JOPES incorporates policies, procedures, personnel, facilities, database and reporting systems and underlying World Wide Military Command and Control System (WWMCCS) Information System (WIS) automated support to provide decision makers and their staffs with an enhanced capacity to plan and execute joint military operations. JOPES is a WWMCCS system that supports actual and exercise joint planning during peacetime and wartime environments.

The primary function of JOPES is the creation and distribution of operations plans (OPLANs) and their supporting data (e.g., Standard Reference Files [SRF]). Military planners use OPLANs for the information required to execute a battle plan for a given or perceived threat. OPLAN updates are distributed over the WWMCCS Information Network (WIN) (a WAN that connects the WWMCCS mainframes together) via files called JOPES transactions. SRF updates are disseminated via tapes.

JOPES was developed many years ago. It suffers the following problems:

- The architecture is closed. There is no easy way to move to open solutions and standards.
- JOPES transactions are unique to the JOPES community and work only in a mainframe environment.
- Charater-based user interface.
- JOPES transactions lack conflict detection or resolution. The update paradigm used is "last update wins." This paradigm ensures a lack of data synchronization across the network.

In order to alleviate these problems, the Government decided to:

- Replace the mainframe-oriented environment with a modern client/server-oriented environment.
- Replace WWMCCS with GCCS.
- Replace the WIN with the Secret Internet Protocol Router Network (SIPRNET). SIPRNET is an existing government network based upon the industry-standard Transmission Control Protocol (TCP)/Internet Protocol (IP) protocols.
- Replace the existing JOPES data distribution and replication solution with a COTS solution.
- Utilize open system/standards solutions for GCCS.
- Utilize COTS products whenever possible.
- Use the Oracle database product for GCCS JOPES data. The existing IDS-I JOPES database has been replaced with an Oracle database. OPLANs are represented in the ORACLE7 database as a set of rows in all, or nearly all, of the OPLAN-related tables.
- Replace the existing JOPES transaction distribution paradigm with a new paradigm that took advantage of COTS products and open systems/solutions.

SRA was tasked to develop the new JOPES transaction distribution paradigm. Our analysis led to the selection of both Oracle and Sybase's replication products for prototyping. The remainder of this paper discusses these two products and the prototypes we developed.

2. PRODUCT EVALUATION

2.1 GCCS JOPES Replication Requirements

The prototypes had to show how well the data replication software could satisfy the following requirements:

- Replicate GCCS JOPES data

  A. Replicate ORACLE7 data
  B. Route OPLAN objects
  C. Autonomous operation and asynchronous data replication
  D. Preservation of referential integrity
  E. Initial data population
  F. Topological support (mesh, star, hierarchical star topologies).

- Avoid, detect and resolve database update conflicts
- Recover from transaction failures
- Promote application and database independence
- Provide database interoperability
- Provide data replication and management tools
- Verify database synchronization

2.2 Product Analysis

With the requirements identified, SRA engaged in research to identify products that could meet those requirements and found that few can replicate and distribute Oracle data in a distributed environment. Some products required extensive application development while others did not. At a minimum, they had to meet the GCCS JOPES data distribution and replication requirements, which includes the ability to replicate ORACLE7 data. Additionally, we gauged the products against the need for being open and for having the flexibility to integrate future technologies as they emerge.

2.2.1 ORACLE7 Symmetric Replication. Symmetric Replication supports two types of data replication: synchronous, which uses a two-phase commit protocol, and asynchronous, which commits transactions to local databases and forwards them to remote databases. Further, it permits mixing these two replication methods on the same network implying that local databases can be updated synchronously while remote databases across a wide area network (WAN) are being updated asynchronously.

Symmetric Replication provides three flexible asynchronous data replication mechanisms: multiple masters or N-Way master replication, updatable (and read-only) snapshots, and programmable asynchronous Remote Procedure Calls (RPCs). These three data replication

mechanisms can be combined in different configurations to meet specific needs. They also have implications for managing GCCS JOPES Standard Reference Files (SRFs) and OPLANs.

- Multiple master replication is a full table, peer-to-peer replication between master tables. All master tables at all sites are updatable. Changes made to a master table are propagated (event driven -, i.e., each master pushes its transactions to every other master) and applied to all other master tables.

- Updatable and read-only snapshots contain either a full copy of a master table or a subset of the rows in a master table. Updates to updatable snapshots are applied locally and are then pushed to the snapshot's master. The master then propagates the transaction to all other masters. Updates originating from the snapshot's master are pulled down by the snapshot's refresh mechanism. Snapshots use a polling mechanism to retrieve updates from the snapshot master site.

- Oracle's programmable asynchronous RPCs allow developers to build their own custom data replication scheme.

Another strength is Oracle's robust collection of conflict detection and resolution schemes. These schemes include, but are not limited to: latest or earliest time stamp, minimum or maximum values, group or site priority, and versioning. If the standard conflict resolution schemes are insufficient, Oracle provides a means to develop custom schemes within the existing framework.

Overall, Oracle's Symmetric Replication satisfies the GCCS JOPES data distribution and replication requirements. It allows sites to operate autonomously, preserves the referential integrity of its data, provides the topological support necessary to manage both OPLAN and SRF data. It also provides conflict detection and resolution mechanisms that enhance its ability to guarantee database synchronization. Our analysis found Oracle's Symmetric Replication solution to be the best choice to satisfy one segment of the GCCS requirements, OPLAN and SRF replication.

2.2.2 Sybase Replication Server. The Sybase Replication Server supports asynchronous data replication with a primary and secondary concept similar to ORACLE7's updatable snapshot architecture. Sybase's use of this concept makes it better suited than Oracle to support a network architecture configured to reflect the GCCS JOPES business rules that govern OPLAN and SRF distribution and replication. Sybase's primary and secondary relationships among its subscribers lend themselves to a multi-layered hierarchical structure that allows primary sites in one layer to be secondary sites at the next higher level.

Sybase can replicate data among the three main GCCS databases: Sybase, ORACLE7, and Informix. However, like Oracle, Sybase requires modifications to the database to support replication since there is a dependence on triggers and stored procedures to do update notification, routing table maintenance and conflict resolution.

Sybase also compares favorably with the GCCS JOPES data distribution and replication requirements. The choice of one database replication product over the other depends on the priority of the given set of requirements that the product must satisfy. Overall, we rated both products as equals.

2.3 Prototype Evaluations

SRA built proof-of-concept prototypes of the Oracle and Sybase data replication and distribution products to prove the vendor's claims that these products can successfully replicate ORACLE7 data and to satisfy the GCCS requirements.

To do this, SRA established a Systems Engineering Environment (SEE) laboratory (Figure 1) to simulate the GCCS environment as much as possible. Furthermore, the SEE is useful in prototyping database replication solutions using ORACLE7, Sybase and TDS. The SEE consists of three LANs connected to one another via Cisco 4000 routers. These Cisco routers, using control service units/data service units (CSU/DSUs), also allow for WAN simulations.

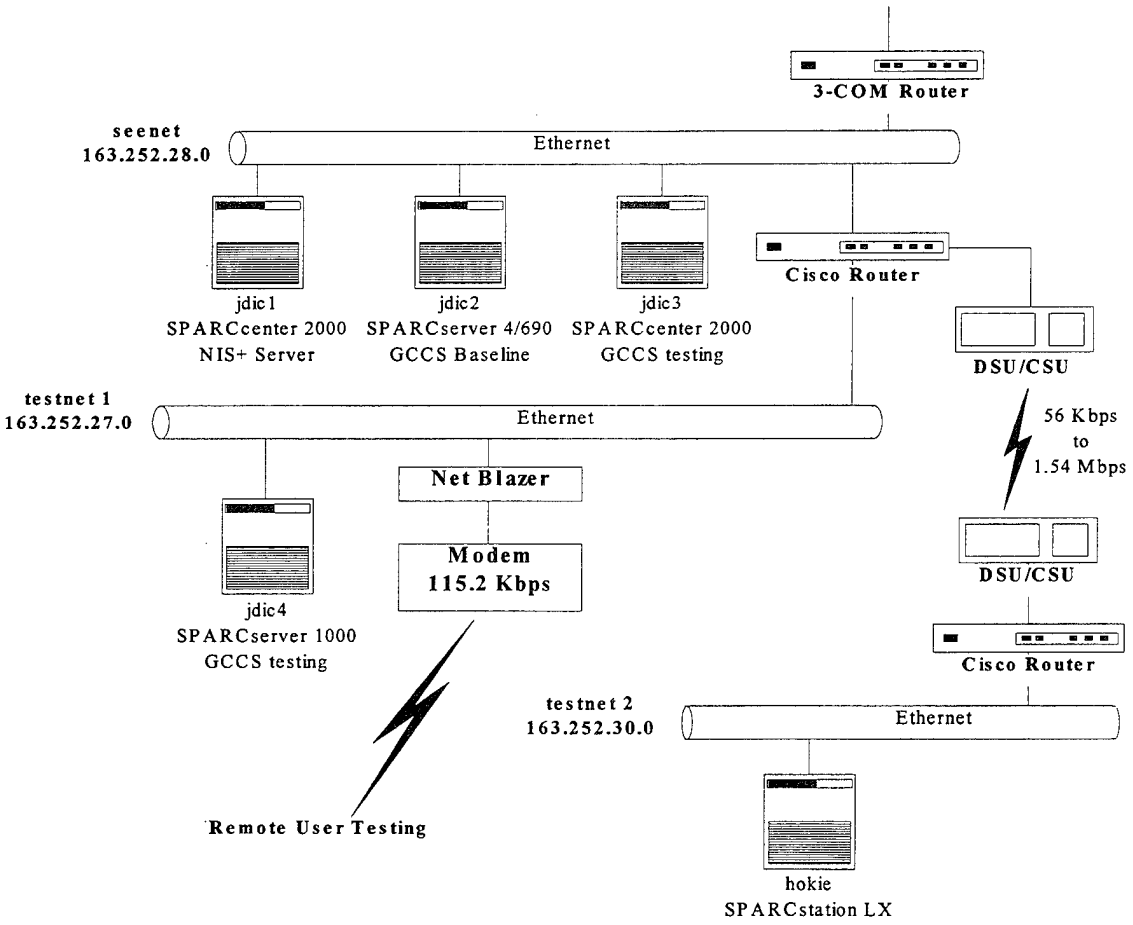


Figure 1: Systems Engineering Environment.

The results of the prototyping efforts were positive; both products replicated ORACLE7 data successfully. They differ in method, and both methods have certain constraints to overcome. These constraints are implementation issues and we discuss them in more detail later in this report.

2.3.1 Oracle Prototype. SRA, with Oracle's data replication consultants, developed two different prototypes. The first prototype shows OPLAN data warehousing by using Oracle's master/updatable snapshot replication features. The second prototype uses Oracle's asynchronous remote procedure call (RPC) replication feature.

2.3.1.1 OPLAN Data Warehousing. With data warehousing, all OPLAN data would be stored at one or more central sites. The remote sites are treated as updatable snapshots refreshed from the master. Figure 2 depicts this multiple master site architecture.

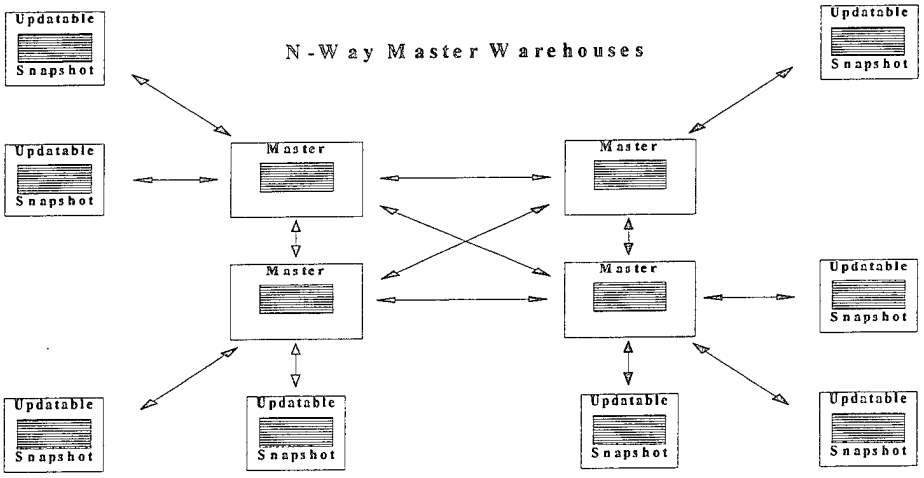


Figure 2: Multiple Replicated Data Warehouses.

In each architecture, the most important aspect is to ensure that snapshot sites receive the appropriate OPLAN updates when they occur. Two methods are available to insure that updates are received: routing columns and routing tables.

The routing column method uses a static route column added to each OPLAN-related table in every GCCS database. This column contains a sub-column for each GCCS site. Custom software will set a site's sub-column with a "true" indicator if the site should receive the row (i.e., OPLAN) updates. The snapshot refresh mechanism reads the master snapshot log to decide if a particular update is applicable to that site. If the update is applicable, the snapshot site executes the appropriate update to synchronize itself with its master site. When a new row is inserted, however, the route column must be updated so that the appropriate sub-columns are set to "true." This routing information is maintained by custom software.

The second method, using a routing table, custom software maintains a routing table at each site. This table is N-Way mastered to all the other route tables. In this manner, the route table at each GCCS site is always current. The table contains the OPLAN ID and the physical database link name, information required to route OPLANs to specific sites. Custom software accesses and modifies the routing table information. The snapshot refresh mechanism reads this routing table and the master snapshot log to decide if a row has been updated that needs to be updated at the snapshot site.

For purposes of prototype development, the static route column was used.

2.3.1.2 OPLAN Update Distribution Using Asynchronous RPCs. This architecture (Figure 3) uses ORACLE7's asynchronous RPC technology to propagate OPLAN changes throughout the GCCS JOPES community. It is similar to the data warehouse architecture in that it uses a routing mechanism built into the ORACLE7 database. That, however, is where the similarity ends. Instead of using master and updatable snapshots, this architecture uses stored procedures that transmit updates to other sites using asynchronous RPCs.

A single routing table (the same routing table as discussed earlier) is created that contains all the routing information for a given OPLAN. This routing table resides on each GCCS server. The routing table at each site is multi-mastered to all of the other route tables at each GCCS site. Thus, the routing table at each site always contains the most current routing information. Custom software must be developed to enable users to maintain the data contained in this routing table.

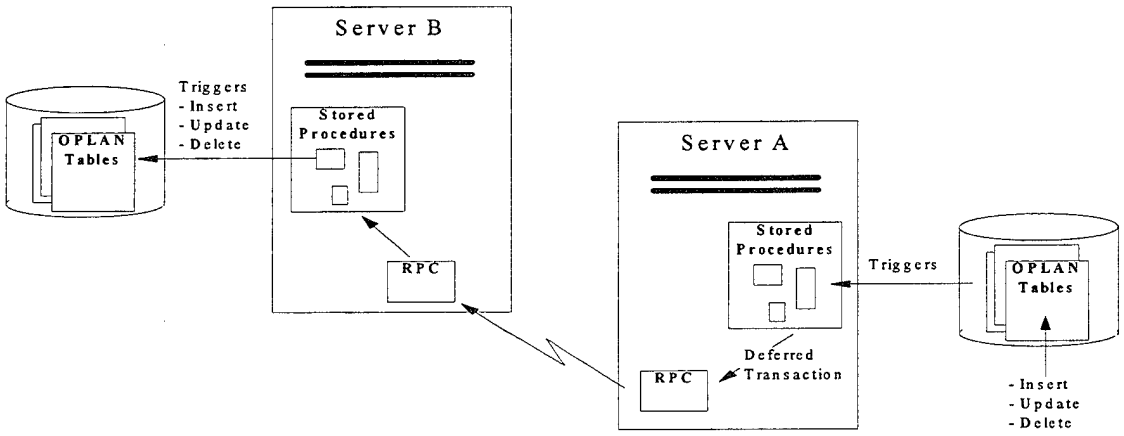


Figure 3: Asynchronous RPCs to Update OPLAN Data.

Both prototypes successfully proved Oracle's data distribution and replication capabilities.

2.3.2 Sybase Prototype. This section will discuss two different solutions for replicating OPLAN data using Sybase's data and OLTM replication features. These solutions were developed with Sybase's replication server engineers and consulting services support. Both solutions were

developed for a peer-to-peer architecture. In the peer-to-peer architecture, the site that originated an OPLAN could become that OPLAN's "owning" site. Any new sites added to the subscription list for an OPLAN would, as a first step, initialize (using either an Oracle stored procedure or the rs_subcmp command) their local copy of the OPLAN data from the "owning" site. Once the materialization process is completed, any outstanding update transactions from other subscribing sites would then be applied to establish an initial baseline for the new site. Every site can make local updates to their copy of the OPLAN. Conflict detection and resolution for a given OPLAN would occur at each site. Figure 4 depicts this architecture.

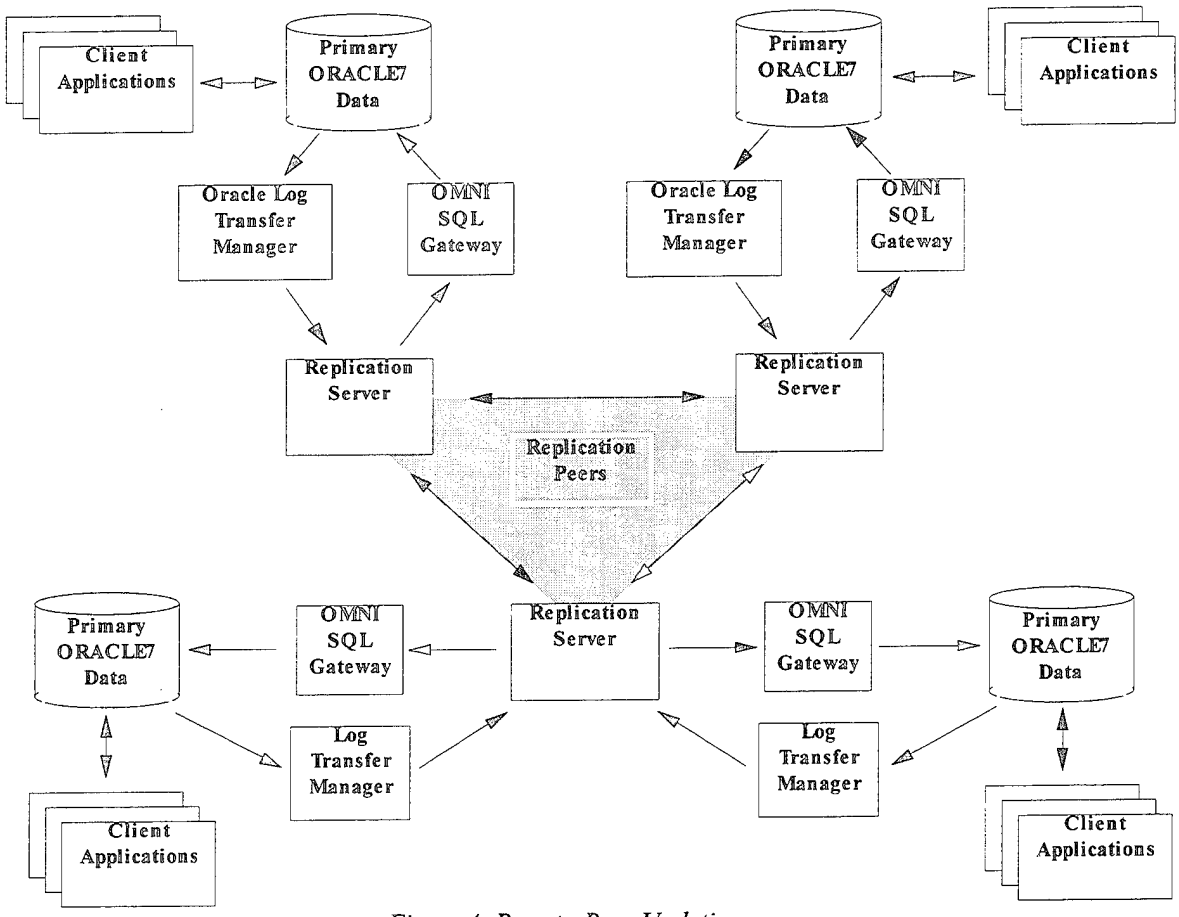


Figure 4: Peer-to-Peer Updating

2.3.2.1 Dynamic Routing of OPLANs Using a Routing Column. To implement this solution, all databases at each GCCS JOPES site will have a static route column added to each OPLAN-related table. This route column will contain a sub-column for each GCCS site. This routing column will map to the Replication Server's rs_address field so that the Replication Server will know where to send updates. Custom software will set the correct column(s) to indicate which sites are to receive any updates.

A route table, maintained by custom software, will need to be maintained at each site. This table will keep track of the sites that are currently subscribing to an OPLAN. This table is required for two reasons:

- Each time a row is inserted into an OPLAN-related table, the route table will be read to decide which site indicators must be set.

- The rs_address stays local to a site. When a new site is added for OPLAN updating, the rs_address mapping at the new site must be added. This mapping information will be derived from the routing table.

Besides the route column or routing table addition, each OPLAN-related table must also add conflict detection/resolution mechanisms to resolve any update conflicts. Resolution of detected conflicts will be accomplished at the OPLAN's primary site. Sybase's function string capability, with ORACLE7 stored procedures and triggers can be used to resolve most conflicts. However, each conflict detected should be written to an error file or log for manual review since every conflict cannot be resolved automatically.

2.3.2.2 Dynamic Subscriptions. This solution uses a concept called dynamic subscriptions. When a site is first designated (via custom software) to receive OPLAN updates, Replication Server subscriptions must be created from the newly added site to the central primary site for a particular OPLAN. The new site, once it has initialized (using either an Oracle stored procedure or the rs_subcmp command) the OPLAN data, is ready to receive OPLAN updates from all other subscribing sites. This dynamic subscription can be accomplished one of two ways:

- All updates to the OPLAN can be quiesced until the new site has completed its materialization process. This ensures that no update transactions were created during the materialization process that the new site would be missing.

- All updates to the OPLAN can be coordinated with the materialization so that the new site gets the initial OPLAN data and any outstanding OPLAN update transactions.

Each OPLAN-related table must add conflict detection/resolution mechanisms to resolve any update conflicts. Resolution of detected conflicts will be accomplished at the OPLAN's primary site. Sybase's function string capability, with ORACLE7 stored procedures and triggers can be used to resolve most conflicts. However, each conflict detected should be written to an error file or log for manual review because every conflict cannot be resolved automatically.

2.4 Product Comparisons

The final step in the product analysis was to compare each product's ability to satisfy the GCCS JOPES data distribution and replication requirements. This comparison is summarized in the

table illustrated in Figure 5. A ✓ shows that the product satisfies the requirement. A ✓- shows that the product does not fully satisfy the requirement or that significant development is required. A ✓+ shows that the product exceeds the requirement or provides enhancements. If the product does not satisfy the requirement, a ✗ is shown. Both Oracle and Sybase performed equally well overall; however, they differed in which requirements they satisfied, and those that they did not.

GCCS JOPES Requirements		Oracle	Sybase
Replication of ORACLE7 data	OPLAN	✓+	✓
	SRF	✓+	✓
Route OPLAN Objects		✓-	✓-
Autonomous Operation and Asynchronous Data Replication		✓	✓
Preservation of Referential Integrity		✓	✗
Initial Data Population		✓	✓
Topological Support	OPLAN	✓	✓+
	SRF	✓+	✓+
Avoid, Detect and Resolve Database Update Conflicts		✓	✓-
Recover From Transaction Failures		✓+	✓
Promote Application Independence		✓	✓
Promote Database Independence		✓-	✓-
Promote Database Interoperability		✗	✓
Provide Data Replication and Management Tools		✓+	✓+
Verifying Database Synchronization	OPLAN	✓	✓
	SRF	✓	✓
Overall Rating		✓	✓

Figure 5: Evaluation Summary.

Assuming there is a need to satisfy all the functional requirement areas listed in the table and that each area is of equal importance and priority, then either the Sybase or Oracle replication product is a good choice. However, this is not so. For example, if the replication of GCCS JOPES data bounded the scope of our selection criteria, then ORACLE7's Symmetric Replication solution would make the best choice at this time. Oracle Corp. offers a powerful and versatile toolkit for

replicating their own data (snapshots, multiple masters, programmable RPCs), and they provide a variety of conflict resolution packages for the replicated database developer and integrator.

If, however, the need to interoperate and replicate data among Oracle Corp., Sybase and Informix databases is a mandatory GCCS COE requirement, then Sybase's Replication Server is a better choice. Sybase has integrated their Log Transfer Managers (LTMs) and Replication Server with their OMNI SQL Gateway, providing an open solution for the interoperability of databases.

There is, however, an additional advantage and disadvantage associated with one replication solution that may offer a deciding factor. The advantage is simplicity: only one mechanism to configure and maintain. The disadvantage may be mediocrity. Using one solution for all applications may force the replication capabilities down to the lowest common denominator. An open environment should promote the use of any products that satisfy the standards, thus allowing each customer to select the best solution for their own particular problems.

Unfortunately, there are no database interoperability standards upon which Sybase, Oracle or Informix has agreed. Sybase uses its own proprietary communications scheme called Open Client and Oracle uses its own proprietary communications scheme called SQL*Net. Until database interoperability standards do exist, SRA recommends that each replication solution be judged on its abilities to satisfy an application's specific needs.

Biography

Ron Harris received a M.S. in Management Information Systems from American University. Currently a senior member of the professional staff at Systems Research and Applications Corporation, Ron is the lead systems engineer developing database distribution and replication prototypes for GCCS. With over 15 years experience in systems development, Ron has led the development of many DoD computer systems.

THE U.S. ARMY CORPS OF ENGINEERS DATA ENCYCLOPEDIA
A FOUNDATION FOR INTEROPERABILITY

BY
STEPHEN VANDIVIER, PRESIDENT
AVANCO INTERNATIONAL, INC.

1. INTRODUCTION

In the modern information age, with our unprecedented access to all forms of data, information is universally viewed as an asset which can be used to gain competitive advantage. Nearly all businesses today have a systematic method for maintaining and evaluating information which controls decisions concerning inventory, products, services, customers, and generally all assets deemed useful to the enterprise. Businesses and the federal government have leveraged their information assets into great productivity gains and cost reductions. New methods of doing business such as "just-in-time" delivery, which reduces static inventories, have evolved due to corporate leveraging of information.

Despite the incredible advantages that organizations have derived from information access, there is also the very real threat of "information overload". Real time data which is inaccessible due to technology (hardware, network) deficiencies, data access methods which are not understood by the organization, and redundant or conflicting information within the enterprise can all lead to frustration by those requiring the information. Such a lack of confidence in the enterprise data management system ultimately leads to lack of confidence in the organization itself.

In any organization where the sheer volume of data is enormous and often overwhelming, there is a true need for a systematic method for managing corporate information. The U.S. Army Corps of Engineers (USACE) has developed an enterprise tool which allows any organization to build a business area model depicting all data elements in the enterprise and their inter-relationships. Known as the USACE Data Encyclopedia this centralized tool stores IDEF data models of business processes within the enterprise. This paper will discuss the concepts, capabilities, and current practical usage of the Encyclopedia within the Corps of Engineers and the DOD and its potential use as a focal point for joint service interoperability and data standardization.

2. THE NEED FOR A DATA REPOSITORY

The trend in business information systems is toward corporate shared knowledge bases. But knowing this is not enough. When you begin a new information system project, how do you know where you stand in the midst of technology evolution. How do you know what the current state-of-the-art is? How do you know whether constantly changing technologies are going to

affect your designs? How do you know where your system designs fit in the hierarchy of designs? The move toward information megacenters and integrated information management environments is accelerating. Missions are rapidly changing, budgets are being slashed, and there is a need to eliminate stovepipe systems and move to modern integrated processing environments. The need to mercilessly eliminate the non-value-added business processes is imperative. Today, you can either change or be changed, have lunch or be lunch. It is a certainty that change will visit your business processes, your information systems, your missions, your capabilities, and your jobs.

A metadata repository can be a key component of survival in this environment because it offers a language for precise, concise, and relevant communication, unifying the project teams, database administrators, business engineers, and application developers as depicted in Figure 1.

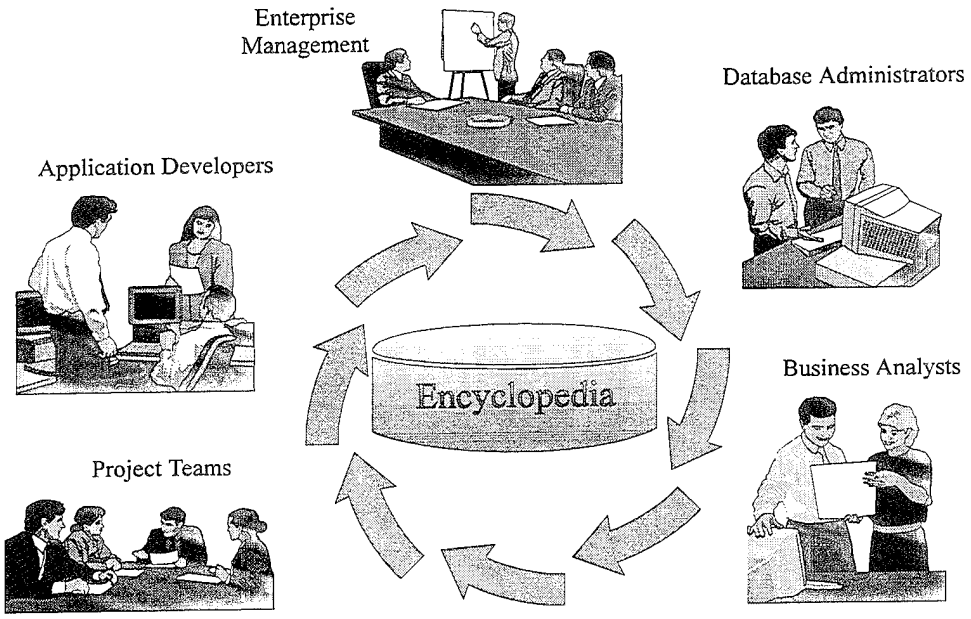


Figure 1.

Information management and software development are closely related issues. The USACE Encyclopedia repository unifies information system concepts such as system development, database implementation strategies, networks, data standardization, DoD standards, DBA practices, support tools, and data integration. Until now there has been little integration of process improvement, CASE, software engineering, database administration, and information management. The repository provides the focal point for the integration and combined application of all these information system elements. It unifies methods, processes, techniques,

tools, and objects, and makes information about these elements readily available to all interested individuals.

3. ENCYCLOPEDIA CONCEPTS

The USACE Encyclopedia was built as a way to manage our response to the information explosion, the technology explosion, and the changing business environment. It is imperative to manage, standardize, understand, and share information so that it can be used as a strategic tool. In order to modernize technologies, it is necessary to constantly monitor and reengineer business processes. The Encyclopedia supports all these goals.

The Encyclopedia is an integrated repository of corporate metadata which include such things as data models, network designs, software designs, and system documentation. Effective in all project development phases, it supports life cycle management of data and information. For software engineering, it facilitates development, maintenance, and technical support of applications. As a resource library, it provides objects and tools to model and integrate information assets.

Just as a library catalog system stores data about books (which themselves contain data), the Encyclopedia stores metadata (data about data) that describes the context and meaning of other data (the data contained in an AIS). Examples of metadata include entity name and definition, data element definition, and data element domain attributes.

Supporting correlations between metadata elements, the Encyclopedia provides an organized automated way to perform:

- Impact Analysis
- Configuration Management
- System Documentation
- Life Cycle Support

4. THE ENTERPRISE MODEL

Information models identify things of importance, such as business rules and data entities, that are needed by the enterprise to achieve mission goals, functions, and strategies. The Encyclopedia uses the IDEF0 and IDEF1X function and Information modeling techniques to transform real world requirements into a logical and physical schema which can be correlated within the tool. Functional requirements are analyzed and transformed into conceptual representations of the business. These conceptual models are further broken down into high level system requirements and the logical database schema. The logical schema is then transformed into a physical database structure. The conceptual and physical models are used in tandem with the system transaction requirements to identify the external views and user interface requirements (system code). All of these functions are supported directly within the

Encyclopedia tool or via direct interfaces between the Encyclopedia and recognized third party tools such as LogicWorks BPWin and ErWin.

As an IDEF Repository, the Data Encyclopedia is a functional representation of an ANSI-defined metadata repository. Its power is derived from its ability to associate items of information that have been collected and stored from three different functional viewpoints, or schemas. This associative capability provides the Encyclopedia user with unparalleled Impact Analysis and Configuration Management capabilities. The functional user can accurately project the impact of proposed changes to a set of procedures, or to components of an automated business application.

The Encyclopedia is based on, and supports, the three schema model as depicted in figure 2. The External, or User-View, schema consists of information concerning Reports, Forms, or entry/retrieval screens that represent the user's interaction with an automated business application. The conceptual schema (or conceptual model) provides the system design framework. The Conceptual viewpoint, or schema, consists of Activity (IDEF0) models and Data (IDEF1X) models. These models can be constructed by utilizing the appropriate entry screens of the Repository, or can be uploaded from a variety of PC-based IDEF modeling tools. The Internal, or Physical schema, consists of information about the file structures (tables, columns, fields, record relationships) that form the foundation of an automated business application. The three-schema model is extended by the Geotechnical Architecture, a repository entity for the collection and organization of data about program procedures, application information, computer system architectures, and installation sites.

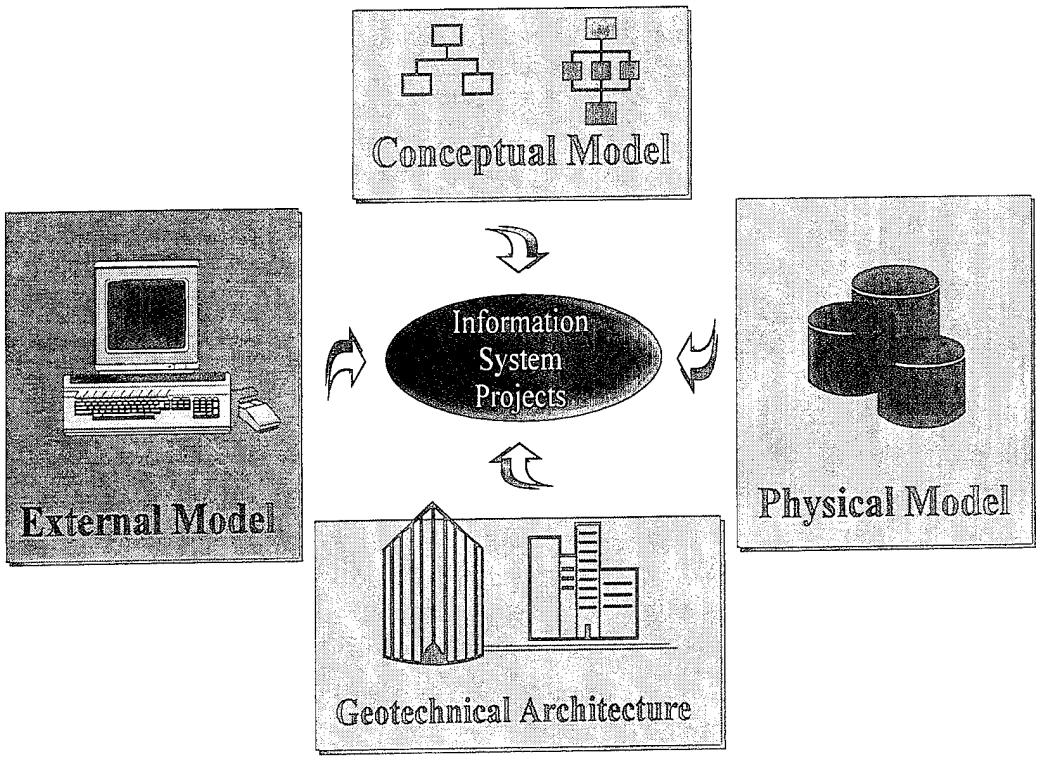


Figure 2.
Three Schema Model with Geotechnical Architecture.

The Encyclopedia Conceptual Schema includes:

- Information Models
- Function Models
- Process Flow Models
- Business Rules

The Physical Model Schema describes the physical structure of the database to include:

- Relational database structures
- Network database structures
- Heirarchical database structures
- Conventional record structures

The External metadata view includes:

- User screens
- Reports
- Paper Forms
- Application descriptions
- Application interface descriptions
- Program Procedure descriptions
- Program interface descriptions

The Geographic-Technical architecture metadata includes:

- Enterprise installation sites
- Computer System Architecture
- Communications
- Operating System(s)
- Proponents and POCs

The Encyclopedia provides a mechanism to establish logical mappings between schema (logical and physical) and corporate metadata, so that the impact of proposed changes can be seen throughout the system. When divided into understandable components, as in the three-schema architecture, information can be better managed and engineered. Through the use of correlations and cross schema mapping, the tool facilitates Impact Analysis, Configuration Management, System Documentation, and Life Cycle Development Support.

The following correlation reports are currently supported in the Encyclopedia:

- Data Entity to Activity
- Physical Field to Data Element
- External Field to Data Element
- External Field to Physical Field

- Physical Field to Program Procedure
- External View to Transaction Requirements
- External View to Program Procedure
- Program Procedure to Computer System Architecture
- Program Procedure to Transaction Requirements
- Business Rules Report

5. ENCYCLOPEDIA PRACTICAL USAGE

How are these reports useful? Imagine that a new congressional action has passed into law which requires significant new reporting requirements and potential modifications to existing information management systems. How can the USACE Encyclopedia streamline the analysis required for this undertaking? Using correlation reports, application developers and data administrators can first determine what information exists in all current enterprise systems and what physical reports will be affected by the new requirements. They can determine how managers and site personnel actually use current information. In concert with business analysts, administrators and developers can identify if required data entities for the new congressional reports already exist and how the reporting information can be derived. Analysts can determine what affect a required data field length change will have across all systems. Impact analysis can identify what tables will be affected by the addition of new data fields and identify any new primary and foreign keys. Configuration Management reports identify the sites that individual applications are running which will be affected by the change, what program release is the most current, and what programs are executing against specific databases in the enterprise. Documentation reports can be generated to identify relevant components of the system which will require maintenance and who the components are assigned to. All of this is performed via reports and queries that can be triggered at the enterprise level or system or sub-system level.

The Encyclopedia assists project managers developing automated information systems by facilitating:

- Search and retrieval of stored models to eliminate redundant efforts.
- Extraction of existing models to leverage new project starts.
- Upload models from PC CASE tools.
- Validate model correctness.
- Graph or plot interim or final models.
- Integrate phased project models to form the composite project view.

Legacy System Analysis is facilitated by employing the Encyclopedia to:

- Identify and document the relevant Geotechnical environments in which the legacy system operates.
- Map legacy system databases to common information (data) models and their function (activity) model equivalents.

- Re-engineer analyzed systems into a shared operating environment.

The Encyclopedia supports Data Administration and Data Standardization by establishing a stable architecture around which these activities can be consistently applied. Database administrators document and control changes to the components of the database structure via a consistent interface that supports the business process.

The Encyclopedia can identify improvement opportunities and manage the process of change. Alternative business models can be developed, stored in the Encyclopedia, viewed at any point in the development cycle, and updated to reflect increased knowledge and understanding of the business processes. As Figure 3 indicates, the Encyclopedia facilitates full life cycle support and business process improvement.

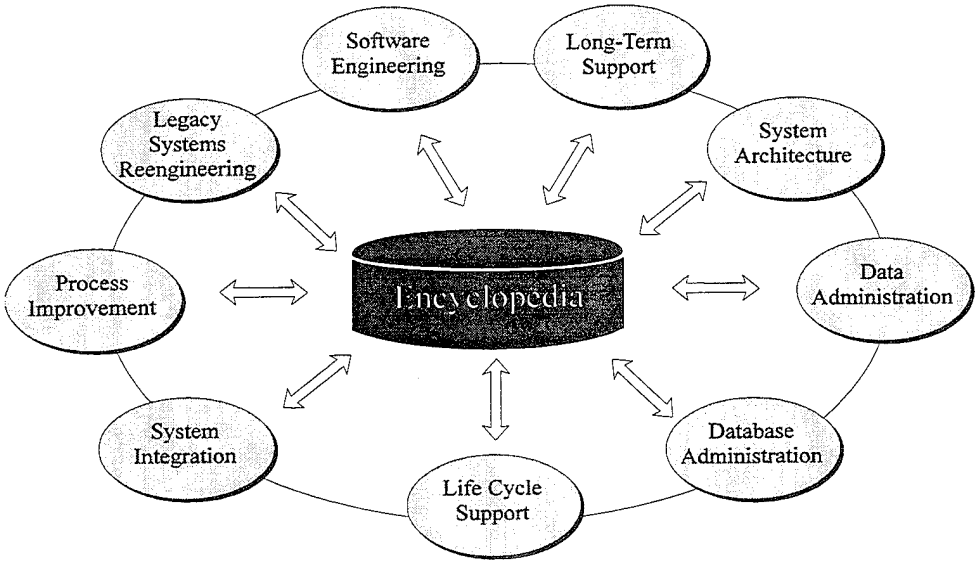


Figure 3.

All objects stored in the Encyclopedia can be called up and viewed for business process improvement analysis, application system enhancement, or new system development. This saves time and money during analysis and design by reusing existing, proven, up-to-date objects; and it helps ensure that system modifications conform to established standards. Reuse of existing information system objects improves system integration and reduces risk.

One of the important Encyclopedia objects is the enterprise data model. The enterprise data model is correlated with the physical and external models, and experienced users can trace requirements from the enterprise data model to the physical model and to screens and reports. Thus the reuse of existing data is encouraged by providing a road map to the data. In this way costly and unnecessary reinvention of manual and automated systems and data is avoided.

The mature and accessible enterprise data model can be used for continuing process analysis and improvement, can be used to communicate information requirements to external organizations, and can be used to communicate and analyze business rules across business areas.

The enterprise data model supports the integration of automated information systems and data sharing. The enterprise data model can help identify the logical point within integrated business processes for single point data entry and single point data storage. The enterprise data model can be used to manage a data dictionary shared across multiple databases that do not duplicate data entities. In other words, it supports normalization of enterprise data across multiple databases.

In a large enterprise, data sharing improves system performance in many ways. Hardware, software, and personnel costs are reduced, and data accuracy, quality, and timeliness are improved. The enterprise data model provides a way to integrate existing information systems and proposed improvements to existing systems. This is accomplished by identifying overlapping business areas, extracting the view of overlapped areas, identifying the associated tables and fields, then redesigning the associated data models to achieve system integration. Other potential benefits of enterprise-wide system integration include reduction of executable code, reduction and standardization of screens and reports, and the enhancement of a uniform user interface.

6. MAINTENANCE

Using the Encyclopedia, all information system components (hardware, software, documentation, databases, procedures, etc.) can be maintained under a single configuration management program. Changes to one component can be analyzed using the Encyclopedia to determine the true impact on the other system components. Thus the encyclopedia can provide technical information used to address shared data issues, negotiate solutions, and implement data structures and definitions which affect corporate operations, procedures, and systems. In short, technical decisions will be based on a thorough multi-level analysis of the entire system. The encyclopedia can be used to support the entire configuration management process, including execution, and monitoring of the reengineering process.

Reengineering projects consist of four phases: ISP, ISPI, STRAP, and System Design. Business reengineering activities begin with the development of an Information systems Plan or (ISP), a document that relates information requirements to missions, goals, and objectives. Next, the Information System Planning Implementation (ISPI) study defines the details of project implementation, including analysis and design. Then, a series of Structured Requirements Analysis Plans (STRAPS), identify business improvement opportunities, AS-IS and TO-BE IDEF0 activity models, and AS-IS and TO-BE IDEF1X models. Finally, new designs are produced of fourth generation, relational, real-time, enterprise-wide systems. As systems become ready for deployment, their detailed, fully-attributed conceptual data models are integrated with the enterprise data model for the final time.

The Corporate Information Management Structure requires the integration of functional Process Improvement, Data Administration, and software Engineering. These goals are achieved using

the Encyclopedia. Methods, processes, techniques, tools, objects, and users can be overlaid on the corporate Information Management Structure, establishing a clear picture of every information engineering project, forming a multi-layered representation of the current state of information engineering, and generating a mechanism for relating, stratifying, and traversing the diverse entities of the Encyclopedia. With this holistic model, the information engineer can develop a project template to guide the initiation, integration, and evolution of an information system project.

7. THE ENCYCLOPEDIA AND INTEROPERABILITY

As the world's largest Construction and engineering organization, the Corps of Engineers has conceived and used the USACE Encyclopedia effectively for enterprise business reengineering efforts and software and data integration. The insight and effort of the developers of the Encyclopedia was rewarded in 1991, by the awarding of the prestigious DoD Golden Nugget Award for excellence in Information Management. The DoD has entered into an agreement with USACE to acquire operating rights over a version of their repository. This repository "clone" is used to analyze, integrate, share, and store IDEF models until the broader scoped version, Defense Information Repository System (DIRS), can be defined and created.

Once an enterprise with many disparate proprietary information systems and redundant information assets, the Corps of Engineers has solved its interoperability problems by standardizing on the USACE Encyclopedia and the Command Data Model which resides in its repository. The Data Encyclopedia tool facilitates the ability to analyze and "grow" a cohesive Command Data Model by providing:

- Integration for all unit models;
- A framework for data administration and data standards;
- Support for the full life-cycle of automated applications;
- A configuration management tool for deployed, distributed applications.

Use of the Encyclopedia within the Corps has resulted in a dramatic culture change within the USACE Information Systems community. Where data and information was once treated as proprietary in the legacy system environment, it is now treated as a shared corporate resource which can benefit all elements within the enterprise. The job of reengineering the organization is facilitated by a tool that allows continuous automated identification, analysis, and implementation of business processes which in turn lead to new and better business practices for the entire organization.

Mr. Stephen Vandivier
AVANCO International, Inc.
7915 Jones Branch Drive, Suite 2B11
McLean, Virginia 22102
Phone: (703) 749-7749
Fax:(703)749-1866
E-Mail:STEVEV@RMF41.USACE.ARMY.MIL

Mr. Vandivier is President of Avanco International, Inc. a small business specializing in business process reengineering, relational database design, and client-server system development. A 1979 graduate of the University of Virginia, he has served the Corps of Engineers and the DOD as a software analyst and Project Manager designing and developing large scale government management and financial systems. His firm of thirty-five software specialists supports the Office of the Secretary of Defense - Policy Automation Directorate, the U.S. Army Corps of Engineers, the Ballistic Missile Defense Organization, Defense Information Systems Agency, and various commercial clients.

Document Management and Production with Relational Database Management Systems

Carter M. Glass
Rapid Systems Solutions, Inc.
8850 Stanford Blvd., Suite 4000
Columbia, MD 21045

1. Introduction

One of the chief challenges facing successful systems integration is the truly seamless integration of disparate applications. All too often data is in the "wrong format." This problem loomed large in the early days of computer word processors; documents written in Multi-Mate, for example, couldn't be opened in Word Star. Indeed, many companies specialized in document conversions, and developed proprietary software to convert documents from one application to another. Nowadays, this problem has been solved by the marketplace. Vendors are anxious for users to adopt their product, so most word processors have filters built-in to convert documents from a competitor's products to their own. The same is true for spread-sheets and PC databases. While this problem has largely been eliminated among applications of the same type, it is still a stumbling block in sharing information among systems which include many different types of applications.

Now users want to take data generated by one type of software product and view or manipulate it with another. For example, a user might want use a spreadsheet to view information stored in a document, or automatically produce letters from a relational database. Also, users often want to view the same information in different formats. Too often systems are constructed to support one explicit set of requirements such as displaying ASCII text, only to discover that the users are now clamoring for hyper-text.

Fortunately, system designers have three powerful tools that make the management and production of documents easy and solves the problems discussed above. These tools are*: mark-up languages, interchange formats*, and *relational database management systems*. This paper describes each of these tools, and how they can be combined to make it easy for software applications to produce documents. All the examples provided use the SQL Server relational database from Sybase and FrameMaker 4.0, a documentation application from Frame Technology. Although this discussion focuses on two specific products, the methods described can be easily adapted to any environment with similar requirements. These are truly general-purpose techniques.

2. Representation of Textual Information

Computers store text as characters and formatting information. The characters consist of the internal character set representation (such as hexadecimal '0A' for the letter English letter 'Z') and the formatting information consists of tokens describing how the characters are to ultimately be presented (such as <B>HELLO WORD<B> to indicate a bold-face font). Hence a document can be represented as simple ASCII text with no formatting at all, or as a page similar to this one, with different fonts, font weights and page positions. The challenge in converting is replacing one set of formatting codes with another. Each application has a unique method for storing and processing formatting information. This is why documents written in one brand of word processor cannot be opened or viewed with another unless a conversion filter is applied, and once the document has been converted, it cannot be saved again in its old format. Inevitably some

formatting information does not convert correctly, leading to the familiar sight of users manually repaginating documents and fixing fonts.

Mark Up Languages

One of the most exciting recent developments in information systems is the growth of the world-wide web (WWW) and hyper-text. The primary reason for the success of the WWW and hyper-text is the idea of a *mark-up language*, specifically Hyper-Text Mark-up Language or HTML. The key feature of a mark-up language is that it stores text as *content* and *structure*; the ultimate appearance of the text is application dependent. All the formatting is performed by the client software. This is a simple, but powerful concept. Consider the HTML statements below:

```
<Title>This is a Title</Title>
Hello AFCEA<P>
```

This collection of statements direct the client application to display the words "This is a Title" in the "title" format, and the phrase "Hello AFCEA" in the regular "paragraph" format. How these appear is entirely determined by the application displaying the text. Thus, some applications, such as a web browser, might display:

This is a Title
Hello AFCEA

Another application might display the text this way:

This is a Title
HELLO AFCEA

And a character-based application would show:

```
This is a Title
Hello AFCEA
```

Notice however that this information can be displayed by *any application on any system.* HTML is essentially a universal language for representing textual information. Furthermore, this concept can be extended beyond simply displaying text in attractive formats. *Any information in a mark-up language can be used by any application that understands the structure tags.* It is just as easy to import this information into spread-sheet cells or rows in a relational database as it is to display it in a text browser. All that is needed is a filter that can translate mark-up language tokens into some form (such as word processor macros, Unix text processing commands or delimited text) that can be read by the receiving application.

Another advantage of representing documents separately as content and structure is that text searching becomes elementary and powerful. In a library of HTML documents just titles or footnotes or headings can be searched for a specific word. This capability is beyond the reach of most word processing systems.

Interchange Formats

Interchange formats are similar to mark-up languages. In an interchange format, all information in a document, paragraphs and pie-charts alike, is denoted as commands, and any application that understands the interchange format can transform the statements back into the original text, drawings or formatting. Also, since the commands are ASCII words, any system can read documents represented in interchange format.

Take as an example a document that includes a picture of a rectangle. Each word processing application will represent this rectangle differently. Hence, converting drawings from one application to another is difficult. An interchange format however makes this much easier; it represents this rectangle as statements. For example, the lines below show how to describe a 2" x 1" rectangle with the upper left corner 3" in from the left margin and 1.5" down from the top margin:

```
<Rectangle
      <ShapeRect 3.00"  1.5"  2.0" 1.0">
>
```

These statements is all that is required for any application needs to draw a rectangle. Additional statements could be included to describe the color of the object, the fill pattern, the line weight an any other properties of the rectangle. Using an interchange format, any object, no matter how involved, can be ultimately be reduced to a finite set of directions.

The product FrameMaker 4.0 from Frame Technology Corporation demonstrates the power of interchange formats. One of the useful features of FrameMaker is its ability to produce files in *maker interchange format* or MIF. Any FrameMaker product can build MIF files, so they provide an easy way to share documents among other products in the FrameMaker family. This guarantees portability across programs, computers and operating systems. MIF files also provide a way for earlier versions of FrameMaker to read documents produced by a later one; in addition, documents in MIF can be easily converted to other formats such as HTML (or a competitor's product!)

Overview of MIF

In a MIF file, all components of a document are *objects*. Objects have *properties*; properties have *defaults*, and everything is described with statements. The true power of MIF, though, is that it will provide defaults for any object not specifically described and any missing statements that might be needed. Hence, the basic information about objects is determined only once, and any subsequent document can use the established defaults. In this way complex documents can be constructed from relatively few MIF statements.

A Simple MIF Example

As an example, consider the MIF statements below:

```
<MIFFile 4.0>
#include Memo.defaults
   <Para
      <Paraline
         <String `Hello World'>
       > # end <Paraline>
   > #end <Para>
```

The seven statements above are all that is required to construct a document in FrameMaker complete with the correct paper size, page numbering, font, font weight etc. The MIF filter supplies the established defaults for all objects that are not specifically described; hence the paper size, font face, font size, margins and all other formatting information is automatically applied. MIF files can also use the familiar `#define` and `#include` constructs. *This is a powerful tool* as we will see when we integrate this idea with a relational database management system.

Relational Database Management Systems

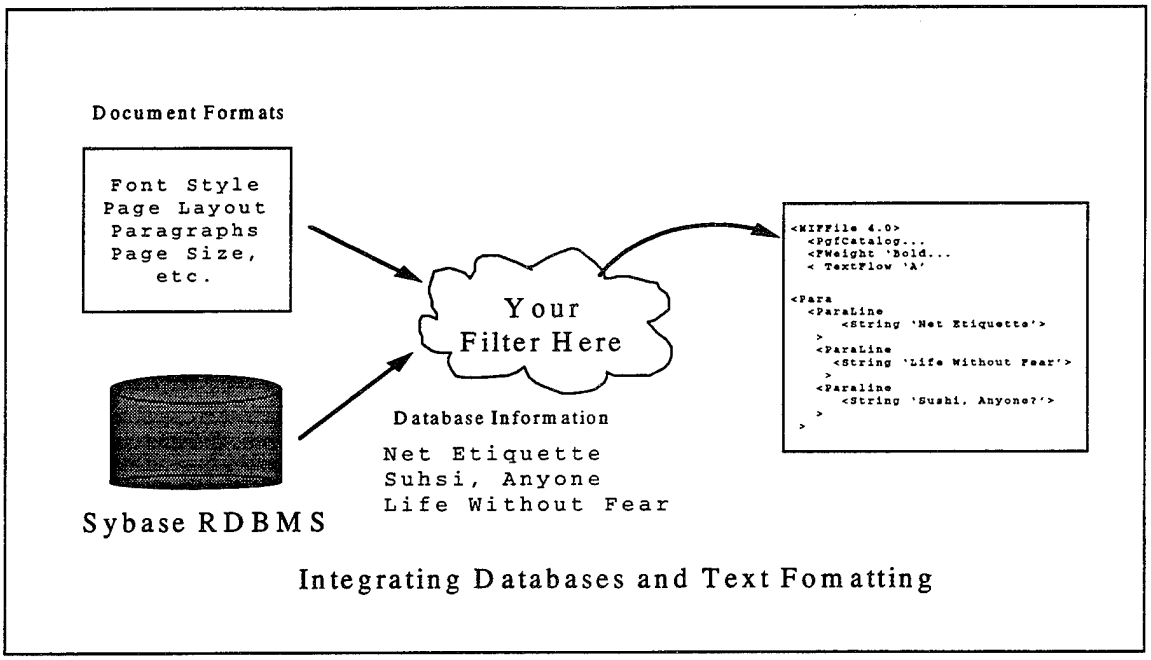
Another tool for text management is a relational database. It has several advantages which make it an excellent tool for managing and producing documents. A database stores information in tables. Tables are constructed from rows and columns, and data stored in tables can be queried, updated and manipulated with a standard language called SQL. Thus, a single database can be used by any number of applications. Each one gets only the information it requires, and it can cast the data in any form. For example, name and address information could be used by a mailing list, a shipping label, and an order-entry program. Since each of these gets the address information from the same (hopefully correct) source, there is no chance of the order-entry program providing a different address than the mailing list. Since the database is not geared to any single use, any new software program that needs address information can get if from the database.

A properly constructed database also eliminates all redundancy from a set of data. This means that each data item is stored only once no matter how many times it might be referenced. An employee personnel record, for instance, might have a last name and so might a payroll record. If each employee in the database was given a unique key, the employee record and the payroll record would each use the key to get the last name. Even if an employee changes their name, their database key will not change. None of the other data rows store the actual name; they only store the key. They will always use the key to retrieve the correct value. Once the information in a database is entered and verified, it is guaranteed to be accurate no matter how it is used. This feature of databases allows them store document "building blocks" as will be shown.

3. Integrating Databases and Text Formatting

The three tools described above -- mark-up languages, interchange formats, and relational databases -- can be combined into an exceptionally productive environment for creating documents. For purposes of example, we have chosen System 10 from Sybase, Incorporated and FrameMaker 4.0 from Frame Technology, Incorporated to illustrate how such a system can work. The basic mechanism is as follows. First, create a template document in FrameMaker. The template provides all of the formatting information: the page size, margins, font and paragraph style, headers and footers, text flows etc. Save the template in MIF format. Next, write a program to retrieve the needed information from the database. This program can be anything -- a stored procedure, a C program, or SQL statements embedded within the application code of a graphical user interface (GUI) tool. Select the data, and write it to an operating system file with tokens similar to a mark-up language. Finally, create a filter to read the both the MIF file and the data file, and use them to construct the final document, which is then available to be opened in FrameMaker.

This technique has several advantages which make it a powerful tool for system integration. The first advantage is that this method completely severs the database from the document application. This has compelling benefits. First, the format of the document can be changed with absolutely no impact on any other part of the system. If the document needs to be created in a new format, the necessary changes are made to the template; no modifications are required to the database or the filter. The database still provides the same data, and the filter uses the changed template to build the document in the new form. This process is illustrated below.



The second advantage is that the database structure, or schema, can be changed with no impact. If the tables are altered as part of normalization or performance tuning, the data retrieval program is updated and the other components of the system require no modifications. In contrast, if the document was produced by a report writer or from within a GUI builder, these would have to be rewritten as well, greatly increasing the amount of maintenance required for each database change. A small database change such as moving a column from one table to another could force the reporting application to be rewritten.

The third advantage with this approach is that the application is no longer limited to its original use. The data file produced by the retrieval program can be used by any program that understands the file format. Hence, a list of data items can be turned into a FrameMaker document, imported into a spread sheet, loaded into another database or moved to another operating system. All that is needed is a filter to construct the final product in the desired format.

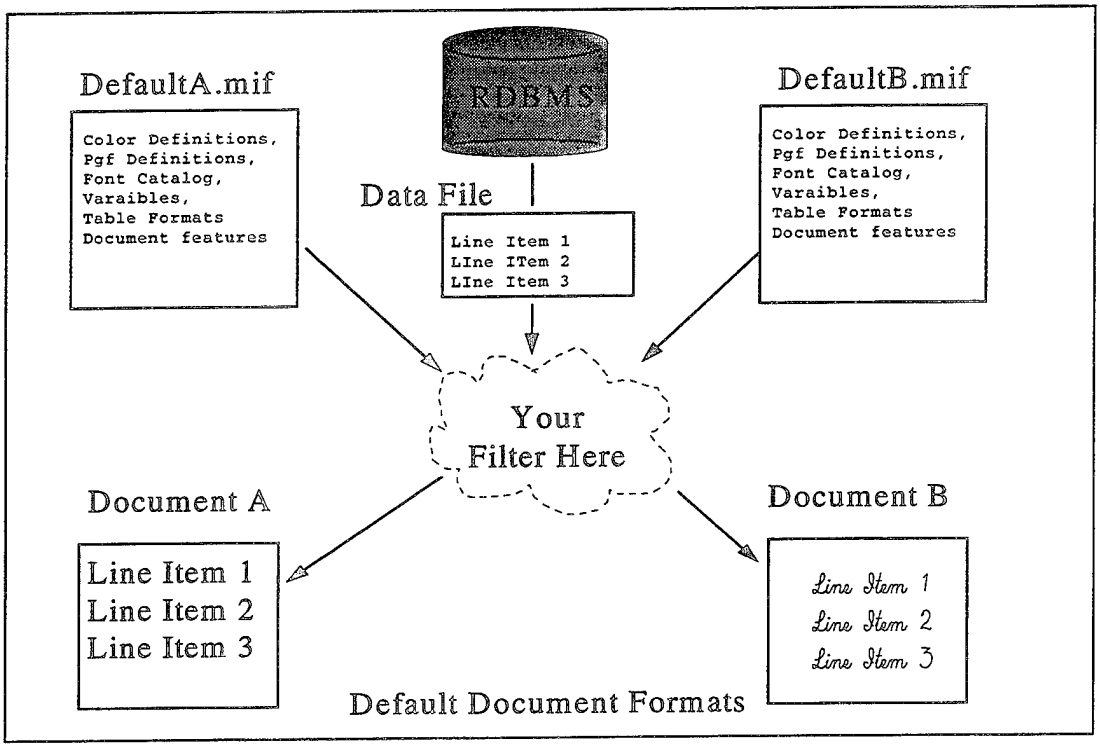
4. Practical Examples from DOD Systems

Several examples of how these techniques have been employed in several DOD systems are provided in the subsequent sections. The first example, called *Default Documents*, describes how different templates can be used to present identical information in two ways. The second example, *Adding Tags*, explains how extending the set of mark-up tokens can provide custom formatting not provided by the original mark-up language. The third example, *Variable Substitution*, illustrates how a standard structure, such as a form, can be completed by substituting data values for place holders. The final example, *Building Blocks*, shows how a database can be used to store text components, and then build a document from those components. Each of these techniques was used in a successful DOD system integration effort.

Example 1: Default Documents

Quite often, systems have a need to present the same information in several forms to meet different reporting requirements. For example, the same list of items from a database might need

to be included in both a letter and an invoice. Usually this necessitates two separate applications -- one to write the letter and one to produce the invoice. If the idea of default documents is applied, the same application can make both reports. First, document templates are created specifying the various formats for the information. Second, a program is written to select the required information from the database. In the third step, filters read the database information and build the documents using the different default formats. The figure shown below depicts one way this might be implemented.



Example 2: Adding Tags

Sometimes the existing structure tags provided by the mark-up language are inadequate. Take as an example a product catalog. The existing structure tags such as `<Title>` `<P>` and `<H1>` provided in mark-up languages are not adequate to describe the structure of a catalog. Some examples of structures that might be included in a catalog are `<Item>` `<Vendor>` and `<Description>`, and these are not provided. The solution is to extend the mark-up language to include the new structures. As an example consider the block of statements below which are a sample catalog entry. New structure tags `<Item>` `<Vendor>` `<Address>` and `<Description>` have been added to the basic set of tags to describe a catalog format.

```
<Item>LEE/G <EndItem>
<Vendor>Emma Emmantors <EndVendor>
<Address>1394 Shrider Road Colorado Springs, CO 80918<EndAddress>
<Description>
A 28VDC Emmanator with Red and Green Lights.
Power cord included
Available in Red, Blue and Green
Please specify color
<EndDescription>
```

With the text in this form, all that is required to produce the final catalog is a filter to apply the appropriate output formats. One filter might display any field marked as an <Item> in bold format, a <Vendor> in plain format, an <Address> as indented text and a <Description> in italic font, like this:

LEE/G
Emma Emmantors
1394 Shrider Road Colorado Springs, CO 80918

A 28VDC Emmanator with Red and Green Lights.
Power cord included
Available in Red, Blue and Green
Please specify color

Another filter might replace each custom token with an HTML token as shown below. In this instance, this document could now be viewed as hyper-text using a web browser.

```
<Title>LEE/G </Title>
<H1>Emma Emmantors <H1>
<H1>1394 Shrider Road Colorado Springs, CO 80918<H1>
A 28VDC Emmanator with Red and Green Lights.
Power cord included
Available in Red, Blue and Green
Please specify color
<P>
```

Finally, a third filter might strip out all tokens so that the catalog could be shown as an ASCII text file. Any new format simply requires a new filter. Notice, also, that if a particular output requires some new information fields, all that is necessary is to create a distinct token for the new data fields and to put the appropriate processing into a new filter. Filters are always written to ignore any tokens that are not understood, so adding new blocks to the data file has no effect on other filters in the application.

If a new catalog format needed to include the price of each item, new tokens could be created to represent this structure, and the application that requires the prices finds the tokens and includes the costs. Catalog formats that do not require the price ignore the tokens. This provides an easy way to support unforeseen requirements with a small amount of new software development, and it insures that data files are always compatible with both new and old applications.

Another advantage of this approach is that a document stored in the extended character format can be easily searched for information. In this example, it would be simple to search through the catalog for a particular item name or vendor addresses since each structure can be quickly located.

Example 3: Variable Substitution

Another common system requirement is completing forms with database information. Quite often, the database and the forms application are not compatible; the usual work-around to this is to print the database information and hand-write it into the form! An answer to this problem is the principle of variable substitution. As an example, consider the form below:

Name:	
Company:	
Title of Presentation:	

If this form is in a table is an application such as FrameMaker, it cannot be filled in directly from a database such as Sybase. This is because Sybase cannot produce output in FrameMaker format, and FrameMaker is limited on the ways it can import data into tables. However, suppose a master version form was created with variables as place holders as shown below.

Name:	$NAME
Company:	$ADDRESS
Title of Presentation:	$TITLE

This master form is saved in interchange or mark-up language format. The database dumps the information to a file with tokens or delimiters similar to the format shown in the Adding Tokens example above. A filter parses the data file and combines it with the master version of the form. The filter either performs the substitution directly or generates a set of commands to perform them. The substitutions are executed, and the completed file is ready to be opened. For example, the filter might produce a file of macro commands like the one below which could then be executed against the master form.

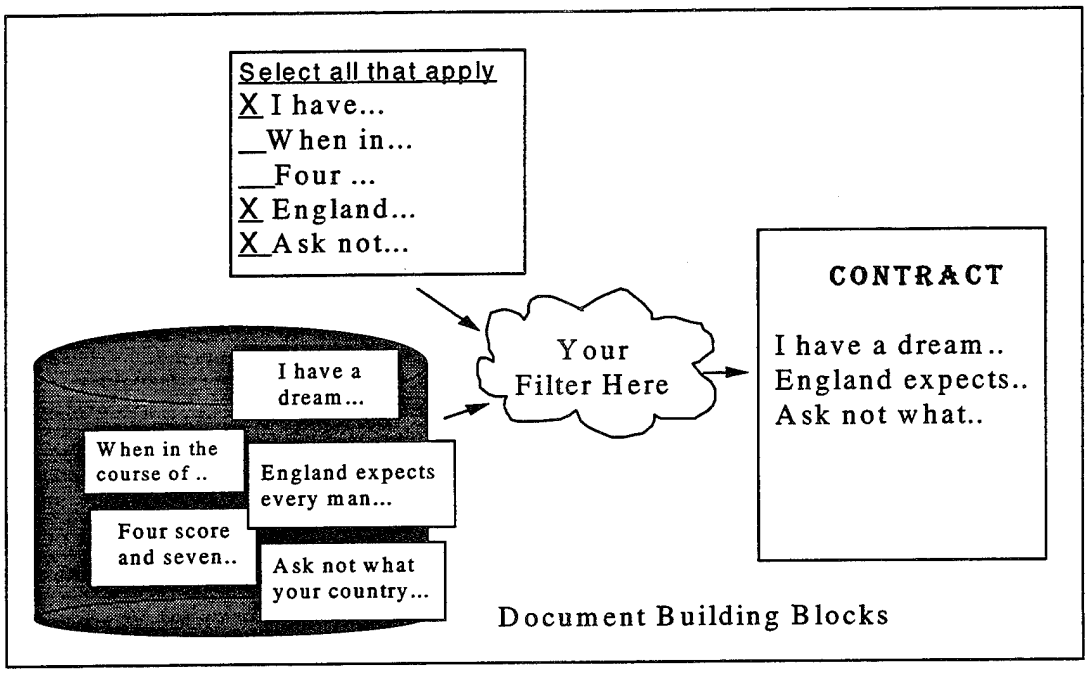
substitute/Carter Glass/$NAME/
substitute/Rapid Systems Solutions/$COMPANY
substitute/Document Management & Production/$TITLE/

Example 4: Building Blocks

A common desire in many applications is building documents from "boiler plates." Contracts are an excellent example of this. A contract usually consists of a collection of standard clauses that are included or excluded as circumstances require. When this is the case, the usual implementation is to create a single "master" copy of the contract which includes the universe of clauses, and each time a contract is created, the master version is copied, and the unwanted clauses are deleted; the remaining clauses are rearranged in the desired order and the contract is printed.

This implementation has several severe drawbacks. First, in order to make sure all contracts include the same language, only one copy of the master contract can exist, and all copies must be made from this version. This is not easy to enforce however, and often users have their own "local" copies of the master contract. This can have harmful consequences if the language in the master copy is changed and users do not update their local copies. Second, customizing a single copy from a master version is labor-intensive and prone to errors. Furthermore, it is difficult to manage individual clauses. If one word in a clause changes, the entire master must be updated. There is no way to view, edit or print clauses separately.

This problem can be solved with the use of building blocks. In this method, the contract is broken down into individual clauses or sentences and stored in a relational database. In the database, each clause is guaranteed to exist only once, and therefore any contract constructed from these clauses will be correct. It is easy to create any possible contract. Users are given a menu listing the possible clauses that can be included. They select the necessary text in the required order. When they are finished, they issue a command to build the document, and the individual clauses are combined into a the final product, and it is displayed for approval. Users can view the document before it is printed. If it is incorrect, they return to the menu and build it again. If some parts of the document need information to be filled in, this is done either by the database as the blocks are selected or when the user sees the print preview. An illustration of how the building blocks method might be implemented is shown below.



5. Conclusion

Simple tools can often be combined in powerful ways. A mark-up language stores all information as structure and content; the ultimate appearance of the information depends entirely upon the application. An interchange format represents any documentation object as sequences of commands. Hence, any application that understands the interchange format commands can recreate any object. A relational database can furnish information in any form and eliminate all data redundancy. Together these tools provide powerful techniques to manage and produce documents. The same data items can be presented in different ways by using default documents. Extending the set of tokens provided in a mark-up language furnishes a way to create new structures and new types of documents. Variable substitution affords a way to insert data into tables and forms, and building blocks are an easy way to construct documents from boiler-plate text. Sybase and FrameMaker are two applications that have be used successfully to implement these solutions, but the techniques are easily extendible to other products.

About the author

Mr Carter M Glass is a Senior Software Engineer at Rapid Systems Solutions, a computer solutions provider in Columbia, Maryland. He has been building database systems for DOD clients for 10 years. He can be contacted at 410-312-0777 or `carter.glass@rssi.com`.

CIM/EI DATA METRICS

Pamela Piper
Defense Information Systems Agency
Center for Software

1. INTRODUCTION

In 1993, the Government Performance Reporting Act (GPRA) was passed. This law requires performance metrics to be developed within all Government agencies, and requires periodic submission of status reports to Congress.

The Department of Defense responded to the requirement for performance measures and management controls in Section 381 of the FY1995 DoD Authorization Act by developing performance metrics in three areas of activity-accelerated implementation of migration systems; establishment of data standards; and process improvement. The metrics that apply to Information Technology are defined in the Corporate Information Management/Enterprise Integration (CIM/EI) Plan, aligned with the goals and objectives of the Plan. The CIM/EI Plan goal that drives the data metrics is Goal 2 - "Tie DOD together through the use of quality, shared data."

The CIM/EI data metrics are also related to the six goals of the DOD Data Administration Strategic Plan (DASP). The DOD DASP was instituted as the primary annual planning document to address and guide the development, implementation, management and resourcing of DOD Data Administration.There is a direct traceability between Goal 2 of the DASP (Standard Data), Goal 4 of the DASP (Quality Data) and Goal 2 of the CIM/EI plan. In addition, there is a direct relationship between Goal 2 of the DASP and the CIM/EI data metric D1, Number of Approved DOD Data Standards. D1 is discussed in greater detail in Section 2 of this paper.

The relationships among the various regulatory, policy, and guidance documents are shown below (Figure 1).

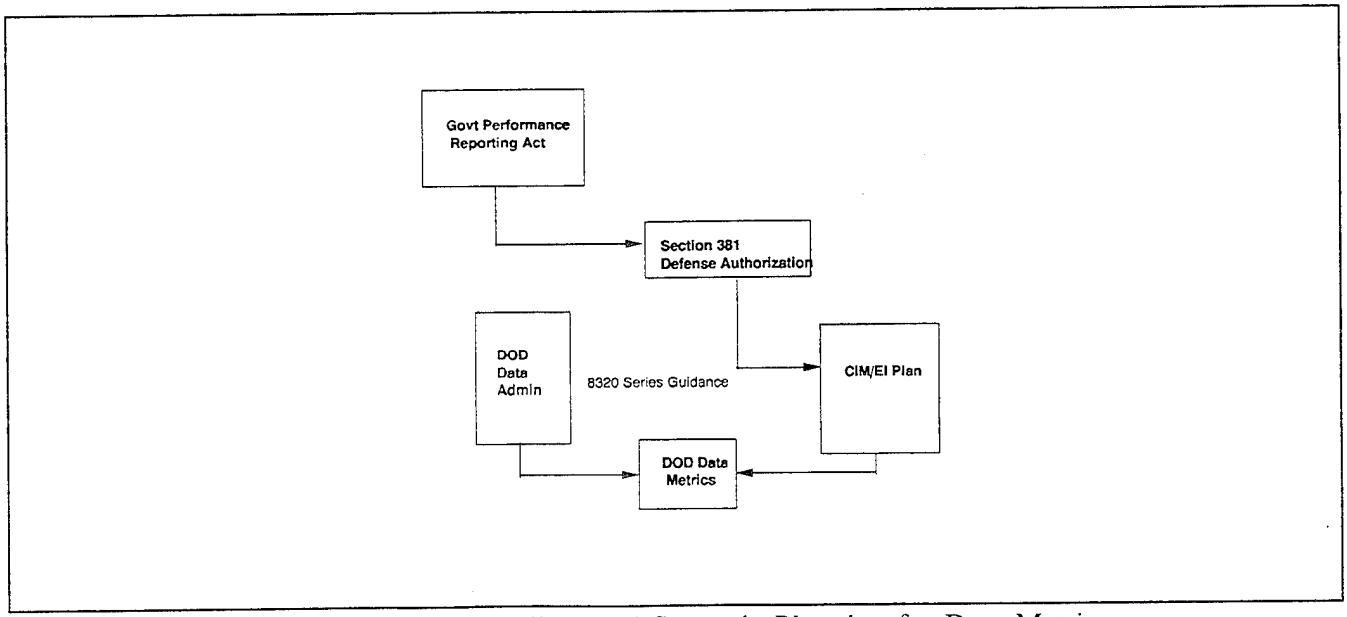


Figure 1 Relationship Between Law, Policy, and Strategic Planning for Data Metrics

2. DATA METRIC D1

Data metric D1 is : "Number of approved data standards". The term "data standards" includes standard data elements, prime words (data entities), and generics (primarily class words). The number of approved data standards are reported for six-month time periods beginning with January of 1994. The DOD aggregate totals are shown below in Figure 2.

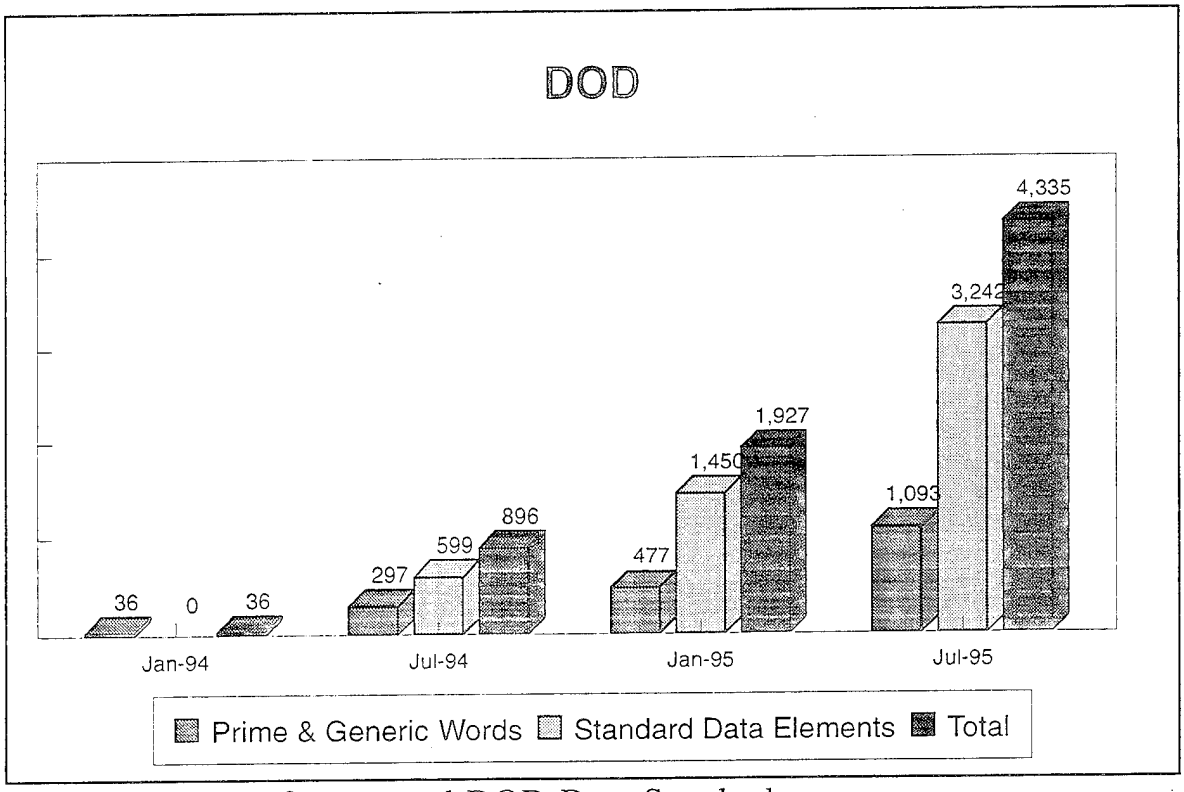


Figure 2 Number of Approved DOD Data Standards

The data source for the counts in each category and time period is the Defense Data Dictionary System (DDDS). The determinant for the time period counts of approved data standards is "authorized-date". "Authorized-date" is a data element within the physical schema of the DDDS, where the values represent the date when the data object was approved as a standard.

3. DATA METRIC D2

The second metric for data, D2, is: "Number and percentage of DOD standard data elements used in migration systems". This metric, when combined with the information in D1, shows progress toward implementing standard data, and, by implication, progress toward shareability.

Understanding this metric requires that we first become familiar with the definitions of "standard", "mapped", and "non-standard data". "Non-standard data" is the set of data elements loaded in DDDS for designated migration systems. By a process of analysis, the functional proponents of the migration system(s) may determine that a data element loaded for a migration system conforms precisely to an existing data standard. In that case, the association between a standard data element and a migration data element may be explicitly declared within DDDS through the use of the "Develop Data Element" menu. Migration data elements that have been identified as conforming to existing DOD data standards are counted as "Standard" in this metric.

The definition of "mapped" data is the set of migration data elements that has been identified as equivalent to an existing DOD data standard, but not yet fully compliant. The test of full compliance is whether an element may be directly shared among multiple functional users in its current state without translation, reformatting, or redefinition.

Non-standard data is all data loaded for a migration system which is neither "standard" nor "mapped".

The percentage in each category is calculated as a percentage of the total number of registered migration system elements. The total of standard, mapped, and nonstandard data equates to the total number of migration system elements registered. Both percentages and raw numbers are shown for each PSA area; for DOD overall, the percentage contribution toward the DOD aggregate total is shown for each Principal Staff Assistant (PSA) area of authority.

For each PSA area, a ratio of the total number of migration systems loaded in the DDDS to the total number of migration systems selected is also shown, next to the legends that identify each PSA area.

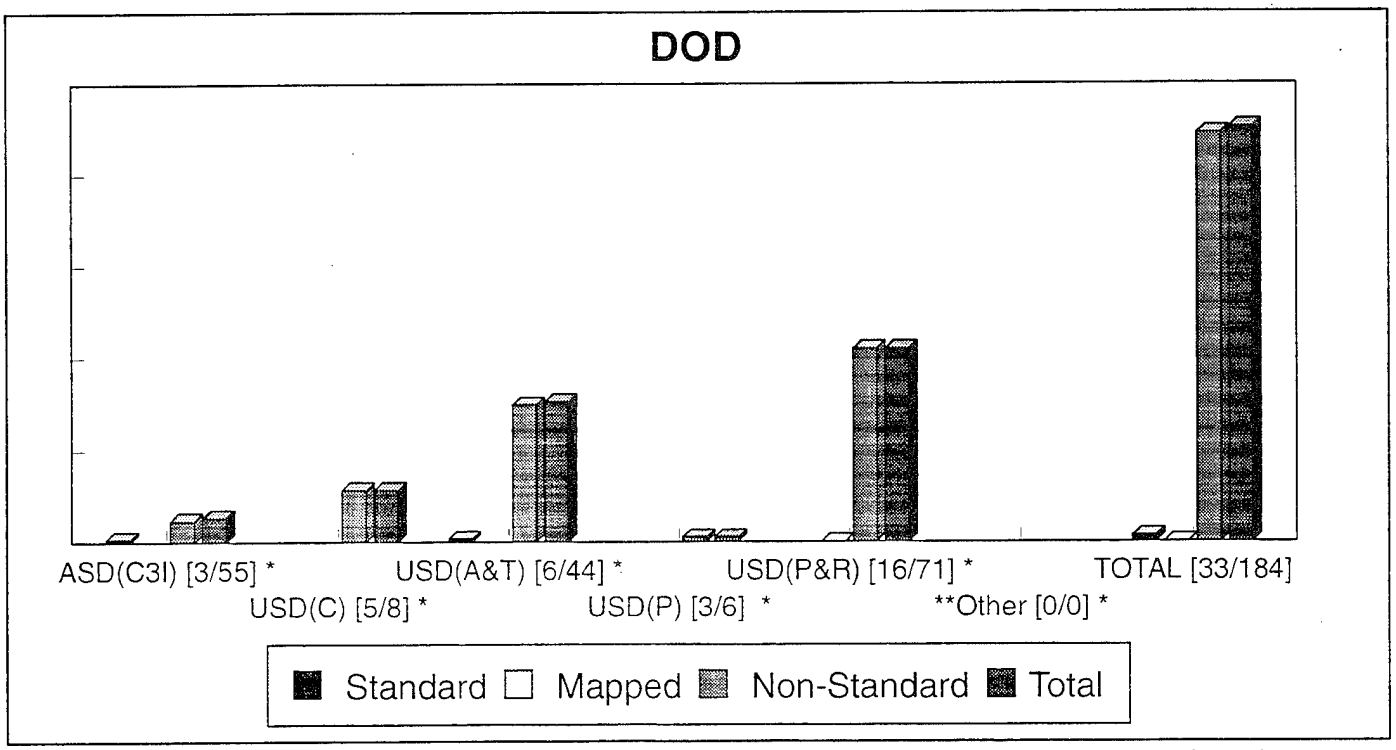


Figure 3 Number and Percentage of DOD Standard Elements Used in DOD Migration Systems

The data source for the counts shown for this metric is the DDDS. The baseline for migration systems is held in the Defense Information System Tool (DIST), under the stewardship of DISA D7.

4. DATA METRIC D3

Metric D3 is: "Number and percentage of DOD migration systems using sharable databases". "Sharable databases" is a term that has been defined in a memo titled "Memorandum for Cross Functional Integration Board Members", released 14 April 1995 by the Deputy Assistant Secretary of Defense for Information Management (DASD(IM)). According to this source, shareable data bases are "those databases which contain data that are shared within or between DoD Components and/or functional areas and are accessible through a common-user access mechanism".

A working draft of D3 is found on the following page, showing both the raw numbers of systems with shareable databases and total migration systems designated in the DIST. These counts are reported for each PSA area and for DOD as a whole. The data source for this metric is the DIST.

5. THE FUTURE OF DATA METRICS

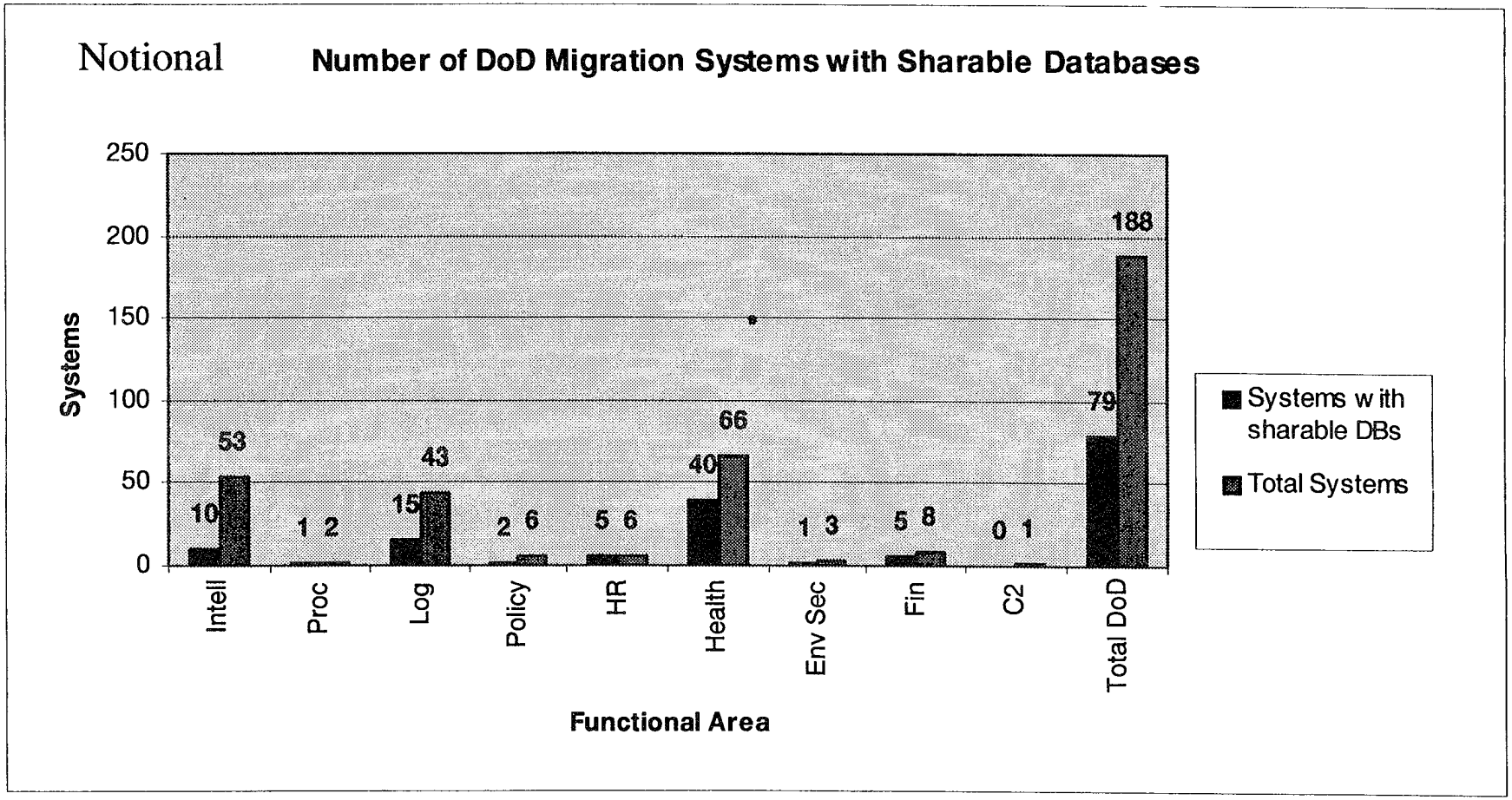
Additional metrics are planned for the areas of migration systems and for data quality. The nature of these new metrics, as well as their form and content, will depend on the management goals and objectives for migration systems and data quality. As we begin to think about developing these new metrics, in the context of the existing measures already in place, it will be important to keep in mind the different approaches to measurement that could be used to define new data metrics. In this section, a taxonomy of metrics is presented as a framework for evaluating the three data metrics that are now in place, and for predicting what form future metrics for data may take.

In Dr. Gerald Britan's paper titled "Measuring Program Performance for Federal Agencies: Issues and Options for performance Indicators", two types of metrics are identified: direct and relational indicators.

5.1 Direct Indicators

Direct indicators of program performance measure what has been achieved in relationship to program objectives [BRI, 8]. The categories are:

Input indicators: measure quantity of resources provided.
Output indicators: measure the quantity of goods and services created.
Outcome indicators: Measure the quantity of direct results that have been achieved as a result of program goods and services.
Impact indicators: Measure the degree to which wider program objectives are being achieved.
Significance indicators: Measure trends with respect to the program goals which



D3 - Number and percentage of DoD migration systems using sharable databases

the program impacts are expected to influence. [BRI, 9-10]

5.2 Relational Indicators

Relational indicators, according to Britan, "measure how well results at one level of the objective tree have been translated into results at the next level." [BRI, 10]. The categories of relational indicators are:

Efficiency Indicators: measure the ratio of inputs needed to outputs produced.
Effectiveness indicators: measure the ratio of program outputs per unit of program outcome.
Relevance indicators: measure the relationship of program outcome to program impact.
Sustainability indicators: measure the persistence of program benefits over time, particularly after program funding ends. [BRI, 10-11]

6. CONCLUSION

The current CIM/EI data metrics, D1, D2, and D3, may be characterized as direct indicators of program performance. This category of metric may be the most appropriate choice for the earliest stages of measuring program performance. As the CIM/EI plan and the DOD Data Administration program continue to evolve and mature, we may expect a trend toward the use of relational indicators. The new metrics developed for data quality and migration data, in combination with the existing metrics, will focus on the achievement of the program goals expressed in 8320 series guidance and the vision of the CIM/EI plan.

Pam Piper is with the Defense Information Systems Agency's Center for Software, and can be reached by phone (703) 681-2148 and by email at piper1p@cc.ims.disa.mil. She is a member of the DAMA Board of Directors for this year (1995-96), and chair of the DAMA Government Special Interest Group. Within the Department of Defense Center For Software, she serves as action officer for the DoD Data Administration Strategic Plan (DASP). She received her BGS and MS in Information Systems from American University, and is currently doing postgraduate study in mathematics and computer science at the University of Maryland.

UTILITY OF DATA MAPPING

LYNN HENDERSON, DEFENSE INFORMATION SYSTEMS AGENCY
FELIZA KEPLER, DATA NETWORKS CORPORATION

Abstract

Data is an essential element in the Defense Information Infrastructure effort. Sharable and mapped data are products of the analysis and coordination necessary to establish a common or standard understanding of the data. Data sharing is concerned with both data integration and the creation of shared data resources. Sharable data is data that is defined and structured such that it can be accessed, used, and understood by multiple systems and users. Data sharing is tied to the management of data as a corporate asset and the determination of common and unique data requirements that must be supported in the DOD environment.

Data mapping is concerned with the movement of data from legacy systems to the designated migration environment and the mapping of data to the DOD enterprise Data Model and external data representations. Mapped data is data that is correlated such that the definition and meaning of related data concepts are identified and supported.

All DOD organizations are responsible for managing the transition of DOD databases towards a shared database environment which utilizes DOD standard data and supports DOD migration and standard systems. This paper describes various roles of data mapping in the migration path from inefficient data exchange through compatible data to shared and distributed data. While the context for discussion is the DOD Enterprise and its objective to implement the Enterprise Data Model and standards in its information systems, the concepts presented are equally applicable to organizations who are introducing data standards into their information systems, rehosting or integrating databases, or implementing data warehouse solutions.

1. BACKGROUND

In July 1989, a Defense Manpower Review outlined changes to improve Defense capabilities. Additionally, the DOD was both experiencing and forecasting declining budget dollars and the need for improved information systems which were robust and flexible enough to meet the changing (e.g., joint) needs of the Department of Defense (DOD). The Deputy Secretary of Defense announced the Corporate Information Management (CIM) Initiative in October 1989. In January 1991, the Deputy Secretary of Defense approved the plan for CIM DOD-wide.

Corporate Information Management is a strategic management initiative, embodied in policies and programs, implementation guidance, and supporting resources, to help functional managers guide and implement changes to processes, data, and systems across the DOD. The overarching CIM objective is to enable the commanders of military forces and the managers of support activities to achieve the highest effectiveness, agility and efficiency in their operations through the effective use of information.

There are 14 guiding principles for the CIM initiative, one of which is that data will be entered only once. Additionally, there are six broad goals for CIM which are in various stages toward completion:

- Minimize duplication and enhance DOD's information systems.
- Tie DOD together through the use of common, shared data.
- "Reinvent" and reengineer DOD operations.
- Implement a flexible, world-wide computer and communications infrastructure
- Apply Corporate Information Management to integrate Defense Enterprise-wide operations.
- Establish CIM policies and management structure.

In addition to the guiding principles, *A Plan for Corporate Information Management for the Department of Defense* outlined eight strategies to be followed by the DOD to implement the initiative:

- Develop models that document new and existing business methods at DOD.
- Develop data standards.
- Develop and implement a set of cost-effective, common information systems.
- Develop and implement a communications and computing infrastructure based on the principles of open systems architecture and systems transparency, to include - but not be limited to - operating systems, database management, data interchange, network/communications services, and user interfaces.
- Manage expenditures for information, regardless of the technology that is applied.
- Institute a life-cycle management methodology that addresses process models, data models, updated system development and acquisition methodologies.
- Establish measures of information management effectiveness and efficiency.
- Educate DOD personnel on the concepts of corporate information management and the plans to apply it.

In his October 13, 1993 memorandum for Accelerated Implementation of Migration Systems, Data Standards, and Process Improvement, the Deputy Secretary of Defense required *(a) Selection of migration systems within six months, with follow-on DOD-wide transition to the selected migration systems over a period not to exceed three years; and (b) Completion of data standardization within three years by simplifying data standardization procedures, reverse engineering data requirements in approved an proposed migration systems, and adopting standard data previously established by individual functions and Components for DOD-wide use wherever practical.*

Three processes are occurring simultaneously which must smoothly converge to achieve the CIM objective:

- Data standardization is underway with a target [baseline] completion of October 1996. As functional data models are completed, they are integrated into the DOD Enterprise Data Model, and their standards are available for use in migration and target systems and shared databases.

- Transition to migration systems requires movement of both users and data from legacy systems to migration systems. The movement of data is complicated by the size of the problem evidenced by the significant number of legacy systems, users, and sites impacted by the movement, all of which must be time phased. Data-related issues can include such questions as: Where is the data to be located? How is database integrity to be maintained during and after transition? How do distributed applications gain access to the data? How will data exchange between legacy, migration, and external (non-DOD) systems be resolved?

- The ability to transition significant numbers of users to migration applications can require changes to the supporting technical infrastructure (computers and communication) on which the applications are run, and data are stored and communicated). The Technical Architecture Framework for Information Management provides the doctrine for system design, and the evolving Defense Information Infrastructure provides the foundation platforms. (The Defense Information Infrastructure, or DII, includes the Deployed Combined/Joint Task Force information infrastructure, the Sustaining Base information infrastructure, and the Enterprise Infrastructure. The DII also provides interfaces to other sources in the National Information Infrastructure and to U.S. Allies.)

The three-pronged approach to achieving goals of transitioning to a reduced numbers of shared migration systems; universal use of shared, standard data in information systems; and transition to open, interoperable platforms is aggressive, with progress in each of the approaches not necessarily coordinated with the other two. For example, if data elements needed by a migration system are not standardized until after users are transitioned to a migration system or the migration system is moved to a new platform, then the application will later have to be modified to incorporate use of standard data, and modifications to the technical infrastructure may be necessary to accommodate shared databases, different DBMSs and schemas, etc.

2. DATA MAPPING DEFINED

In the context of this paper, data mapping is the creation of a cross reference between data items existing in two or more systems or representations of those systems in models. The cross reference specifies the inherent variability or sameness of data and more importantly, resolves user uncertainty as to the content and meaning.

3. DATA MAPPING APPLICATIONS

Data mapping is generally viewed as a task that should be assigned to a journeyman. In point of fact, it is neither prosaic nor simple. Data mapping reconciles disparate data. Considering as we do that in order to be shareable, the inherent variability of data and the users' uncertainty about the data's content and meaning must be resolved, data mapping emerges as a critically important task that requires considerable thought and analysis. The generally accepted use of data mapping consists of using the enterprise data architecture as the basis for the resolution of data variability and user uncertainty. Data mapping then, cross references all existing data to the enterprise data architecture, thereby effecting shareability. However, there are a number of other applications, which can be viewed as incremental measures during the DOD enterprise integration that can benefit from data mapping.

3.1 Consolidation of Functions and Databases

Migration applications selected in a number of functional areas require services from a legacy application, i.e. an application that was not selected for migration. Usually this legacy application provides services to a number of functional areas and the functional area wisely decides not to undertake the migration and concomitant costs of the entire application. Before deployment of the migration application, therefore, the services from the legacy application are to be folded into the migration application.

The recommended basis for the consolidation of functions between the applications would be a logical data model depicting the data requirements of the migration application. There is almost certainly a physical data design that is mapped to that logical data model. The functions that are to be incorporated from the legacy application are then carefully analyzed to determine what data they operate on in the legacy application. The data so identified in the legacy application are then mapped into the migration application's data model. If required, the migration application's data model must be suitably augmented. A physical database design is developed for the augmentation. When the migration application already has a relational database, this is simply a matter of adding additional tables. For other types of database organization, this augmentation may require the addition of new record types or the addition of new fields to existing records.

A potential complication in using this approach to consolidating functions and databases might be the lack of an existing logical data model depicting the data requirements for the

migration application. The logical data model would then have to be reverse engineered for the migration application and the existing database design mapped to that logical model. A second best approach would be to use the physical model, i.e. the database design. The effectiveness of this second best approach depends almost exclusively on how well the migration application's database designer understands the functional meaning of the individual data elements in the existing database or his/her access to organic resources with that functional understanding.

Because of the aggressive development and deployment schedules targeted for completion by October, 1996, there exists the temptation to skip the mapping of the data required by the legacy application to the migration application's data model and to simply add the tables and/or records to the migration application' database. While expedient, this shortcut is likely to increase the redundancy and therefore the inconsistency of data in the migration application's database. We can not overemphasize the importance of evolving the migration system to the shared data environment essential to DOD enterprise integration.

3.2 Reduction of Legacy System Interfaces

During the deployment of the selected migration systems, their interoperability is generally accomplished using interfaces. The interfaces that need to be built for the functional area's migration application(s) as it is installed to support various user locations are determined by the particular mix of legacy and migration applications encountered at the location. In general, the earlier a functional area deploys, the more interfaces it has to build for the migration application(s) it fields. The costs for developing interfaces can become quite onerous. Having to build scores of short-lived interfaces to legacy systems which will soon be decommissioned is correctly viewed as wasteful by functional areas as they prepare to deploy their migration applications. Every effort to reduce the number of legacy interfaces must therefore be made. Data mapping is an important technique applicable for this purpose.

The collection and documentation of required interfaces has been undertaken by most functional areas. What has typically not been done is to categorize and classify those interfaces so that each of the classes can be addressed in a uniform, and disciplined fashion. The specific class of interfaces we are concerned with are those to a legacy system and those from a legacy system. For each of these classes, we further categorize them as those providing/requiring data or those requesting a service.

The interfaces providing or requiring data are easily addressed. We simply map the data provided or required to the model depicting the logical data requirements for the migration application. If a mapping exists, the interface will be retained and the appropriate extraction or reception of data can be accomplished. Otherwise, the interface is eliminated.

The best approach to interfaces that request or provide a service is a little more difficult to determine. The nature of the service being requested by the legacy system needs to be

characterized. If the service is essentially an operation to update data in the migration system's database and return a status as to the successful completion, this is likely a service that will also be required by a future migration application. The interface should be built as efficiently as possible. Even if this type of interface is rooted in function as opposed to data and perhaps even more so, the data items contained in the interface need to be examined, and mapped to the migration system data model. This step is required to ensure that the status information can be suitably returned.

On the other hand, the service requested could be an operation to reference a piece of information in the migration system's database and to return the information to the legacy system. By mapping the data requested to the logical data model, we are able to determine a consistent way in which this request can be satisfied. This entire class of interfaces can then be replaced by a single facility which accepts the service request to obtain information, obtains and develops the required database from the migration application's database and provides the information back to the user. In effect, the class of interfaces is replaced by a shared data resource.

3.3 Data Loads During Deployment

As a migration system is deployed to various sites, the best source of data from which the migration system's database(s) will initially be loaded has to be determined. The source for the data is likely to be the databases of one or more legacy systems. The preferable source is of course, the databases of one or more migration applications.

The mapping target is the logical data model for the migration application. Existing data from the legacy system which the migration application is replacing is mapped to the logical data model. The resulting cross reference becomes the basis for the design of the data load into the migration applications. It is entirely possible that the data to be loaded needs to be obtained from multiple databases. The problem of determining where to map and where to load the data from is an interesting and complex problem: Among the considerations in selecting the application database from which to cross reference the data and eventually from which to load are:

1) the lifecycle phase in which the legacy system is in: In general, legacy systems which have been developed recently are likely to have logical data models that permit effective data mapping.

2) the existing responsibility and authority to maintain the data: In general, the mapping source selected should be the authoritative source for the instantiation of data in the legacy environment.

3) the predicted currency of the data to be mapped: In general, the mapping should be done from the most active database so that the data used for loading the database of the migration application is the most current.

The migration applications already in the user site are, of course the preferred source. The migration application for another functional area should be the source for all the data for which the other functional area is the authoritative source. The rationale for this selection

is both logical and pragmatic. We note that by definition, the migration systems will be operating longer than any one legacy application, hence it can be said to be early in its lifecycle. For data under its stewardship, the migration application is the authoritative source. Finally, the migration application has only recently been made operational and so the data has not been subject to any decay from continuing operations and is therefore most current. The pragmatic reason is that in future sites, the mapping with the migration applications will not need to be accomplished again and the data load software modules developed here can be reused.

3.4 Development of Data Standards

The ongoing debate within the DOD as to whether data standards should be obtained from models or from the data currently implemented in systems is resolved by data mapping. In the past, many of the data elements proposed for standardization came from data models derived from business and process models. In general, information systems implementing these standards were to have been implemented making the standards real as part of the CIM initiative started in the early 1990s.

With the change in direction late in 1993 to select migration systems and deploy those standard systems DOD wide by October, 1996, many functional areas designated migration systems from the current DOD inventory of application systems instead of the "clean sheet" approach that had originally been envisioned. The net result of this action is that many of the data standards that were proposed from models early in the CIM Program were not implemented. Instead, existing and commercial off the shelf systems were selected and are now slated for DOD wide deployment as the interim standard system for the DOD functional areas. A new round of data standardization proposals are in the works.

The tragedy in this change of direction is that the DOD is not able to accrue any benefits from the data standards previously developed and proposed since at the current time, they are not implemented. Thus, the data standardization is not an avenue for sharing. In the near term, in point of fact, because of the relative stability of the data requirements in the conduct of a business, and therefore in the information systems, the data reflected in those standards are probably implemented. All that is required is to map the data in the selected migration system(s) for that functional area to the standards previously proposed/accepted into the DDRS or depicted in the models.

The mapping or cross referencing of the data existing in the migration systems databases to the models that were derived from business process models would have the effect of decreasing the level of abstraction in the original models and abstracting the operational data requirements. The existing data is also an excellent source for the required metadata in the data element proposal packages. Of course, this data mapping is not simple. It must still be performed by a fairly senior person who understands the functional requirements being addressed by the data and who, from experience has the capability to determine how

physically implemented data will fit in with the logical data model. This would represent a significant investment.

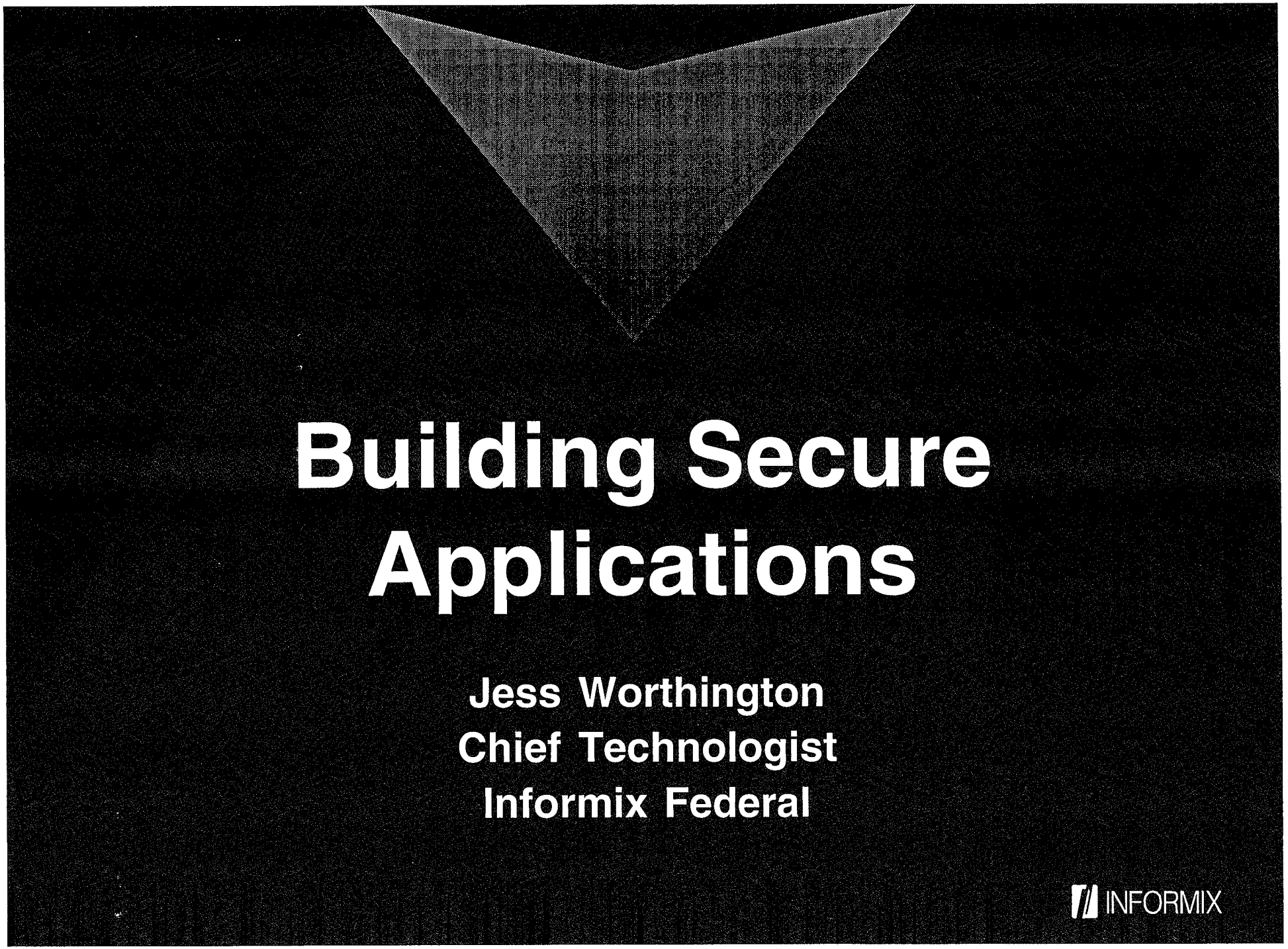
However, the payback is in implemented data standards. The investments already made in data modeling and data standardization can be leveraged upon. The data standards, being mapped to existing data, highlight opportunities for sharing of data across the applications as they exist today, eliminating the protracted period during which the Data Standardization program would appear to have limited near-term returns on investment.

4. SUMMARY AND CONCLUSION

Data mapping is concerned with the establishment and use of data standards and in making the transition from stovepiped legacy systems to an integrated and interoperable system/data environment. This tried and true technique can be used in many different ways to achieve the target data environment.

The transition to migration applications from legacy applications requires data loads of legacy data to migration systems and development of interfaces to other systems that the legacy system interfaced with, as well as to the balance of the legacy system if only part of its functionality transitions. There are database integrity considerations between the legacy and migration applications during, and possibly after, the transition. Additionally, the timing of the availability of standard data models and elements relative to transition from legacy data schemas and elements to migration applications/databases have implications, as well. Data mapping serves a significant role in the transition process whether DOD data are standardized before, during, or after transition to migration systems.

Finally, the DOD cannot afford to migrate all data and must focus on only essential business information. Data mapping can provide added insight to what constitutes essential business information. Mapping existing data to data models which represent the identity and structure of the functional area's data requirements is a key step in data integration that should not be eliminated.


Building Secure Applications
Jess Worthington
Chief Technologist
Informix Federal
INFORMIX

Presentation Overview

- Why do we need a Secure database
- What are Secure systems (> = B1)
- What is a Secure database
- How do I build Secure systems

INFORMIX

Why Do We Need A Secure DBMS?

- To prevent unauthorized compromise of information and performance
- To merge security needs with DBMS functionality
- To reduce the *cost* of data security
- Security level determined by
  - The number of sensitive data categories in maintained Databases
  - The "temptation value" of data itself
  - The "location and technical depth" of those who would be most tempted

INFORMIX

Why Do We Need A Secure DBMS?

- Security is essential for DB Records that require controlled access *and/or* guaranteed integrity of data
- The level of security depends on
  - The level of temptation the data presents to "outsiders"
  - The hardware/software/interconnect configuration
  - The location of "outsiders" most tempted
  - The level of technical competence of those tempted
  - The true intent of those most tempted
  - The full downstream costs of losing data confideniality *or* data integrity/accuracy

INFORMIX

What Is A Secure System?

- Government levels of assurance
  - C2 — Basically UNIX with auditing
  - B1 — Mandatory labeling
  - B2 — Minimal TCB, covert channel analysis

INFORMIX

What Is A Secure System?

Definition

A set of applications and storage mechanisms that, by themselves, will not disclose sensitive information contained in the system.

INFORMIX

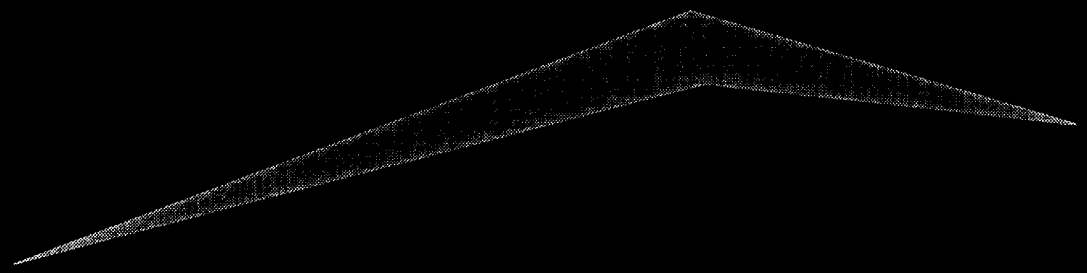
Secure Terminology

- SSO — System Security Officer
- SA — System Administrator
- DBSA — Database System Administrator
- DBSSO — Database System Security Officer
- AAO — Audit Analysis Officer

INFORMIX

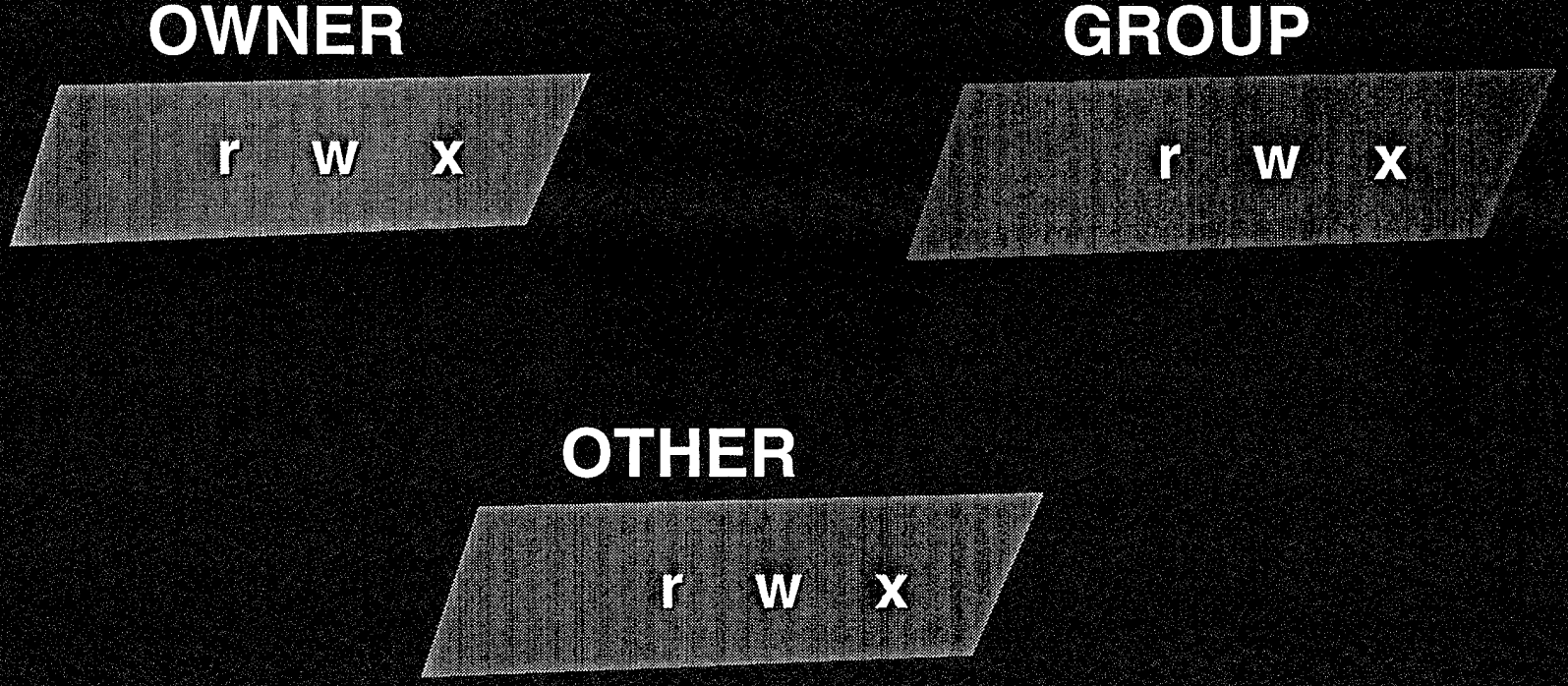
DAC Definition

That access that the owner of on object grants to another user.



INFORMIX

UNIX DAC



INFORMIX

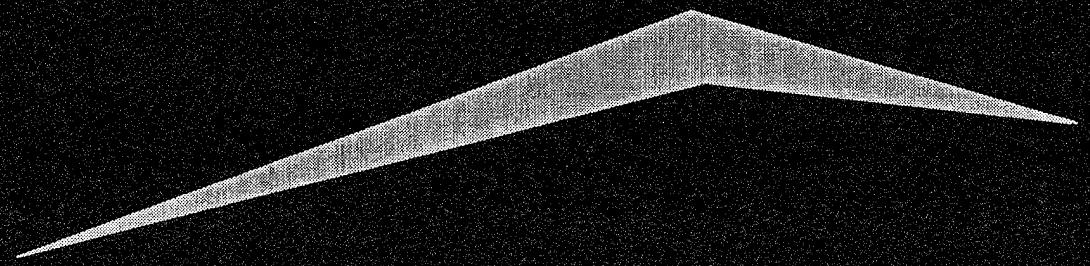
SQL DAC

- GRANT SELECT, INSERT, UPDATE, DELETE ON *table* TO *user*
- REVOKE DELETE ON *table* FROM *user*

INFORMIX

MAC Definition

MAC is access based on user security clearance level within a specific domain and is system enforced.



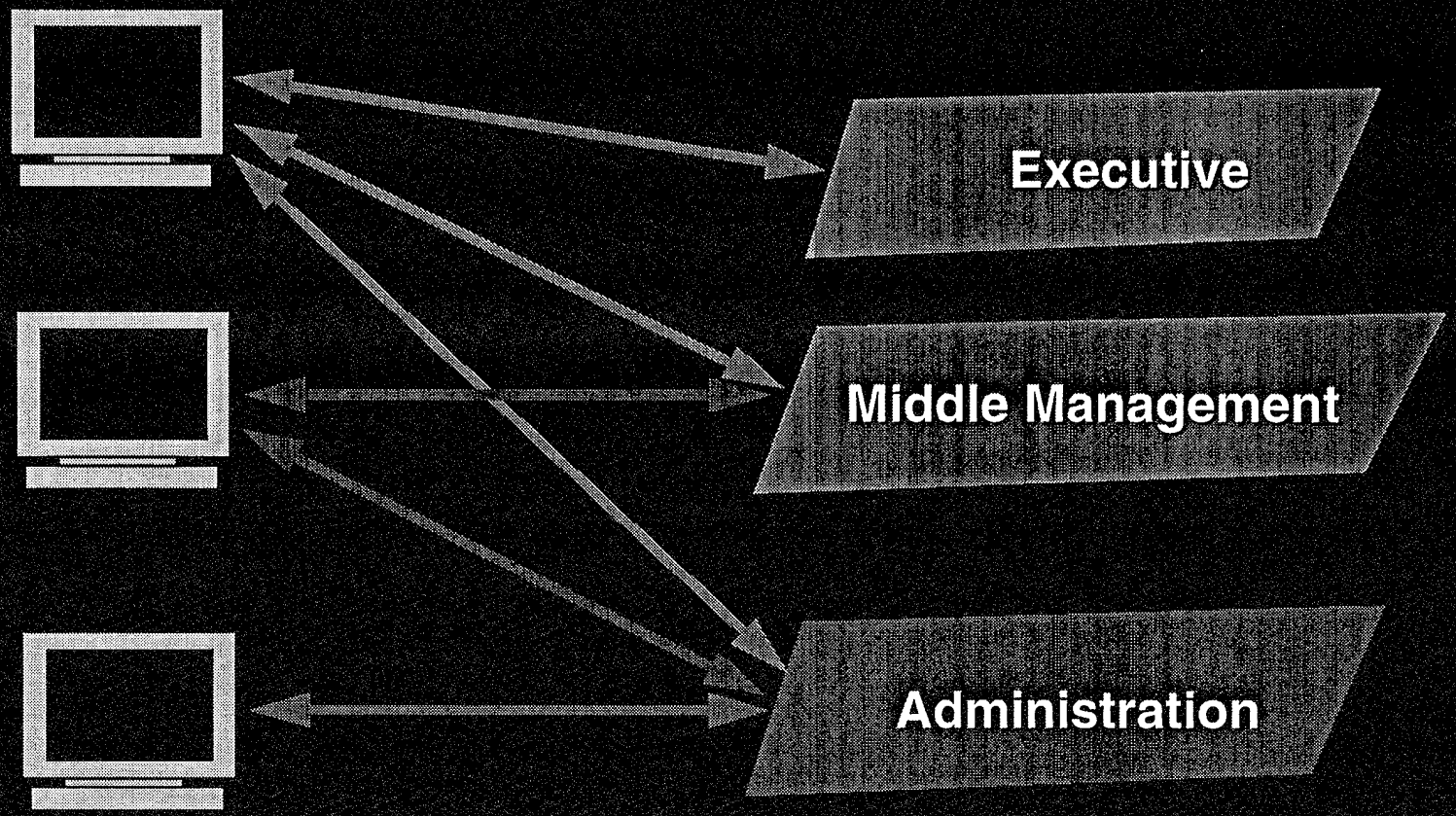
INFORMIX

Secure Terminology

- DAC — Discretionary Access Control
- MAC — Mandatory Access Control
- TCB — Trusted Computing Base
- ASSURANCE — Validated Level of Trust

INFORMIX

Mandatory Access Control



INFORMIX

UNIX vs Trusted UNIX Differences

- ☆ Superuser can do anything
- ☆ DAC file access control only
- ☆ System oriented auditing
- ☆ Single level directory structure
- ☆ Superuser function split into roles — no one with total access
- ☆ DAC and MAC file access control
- ☆ User-oriented auditing
- ☆ Multilevel directory structure

INFORMIX

OnLine vs. Secure Product Differences

OnLine

- ★ All data stored at a single level
- ★ Single DBA role
- ★ OS provides user verification
- ★ Discretionary access control (DAC) only

OnLine/Secure

- ★ Data stored at multiple levels
- ★ Multiple roles for DBA, DBSO, SSO, Audit
- ★ OS provides user verification and security level
- ★ DAC + mandatory access control (MAC)

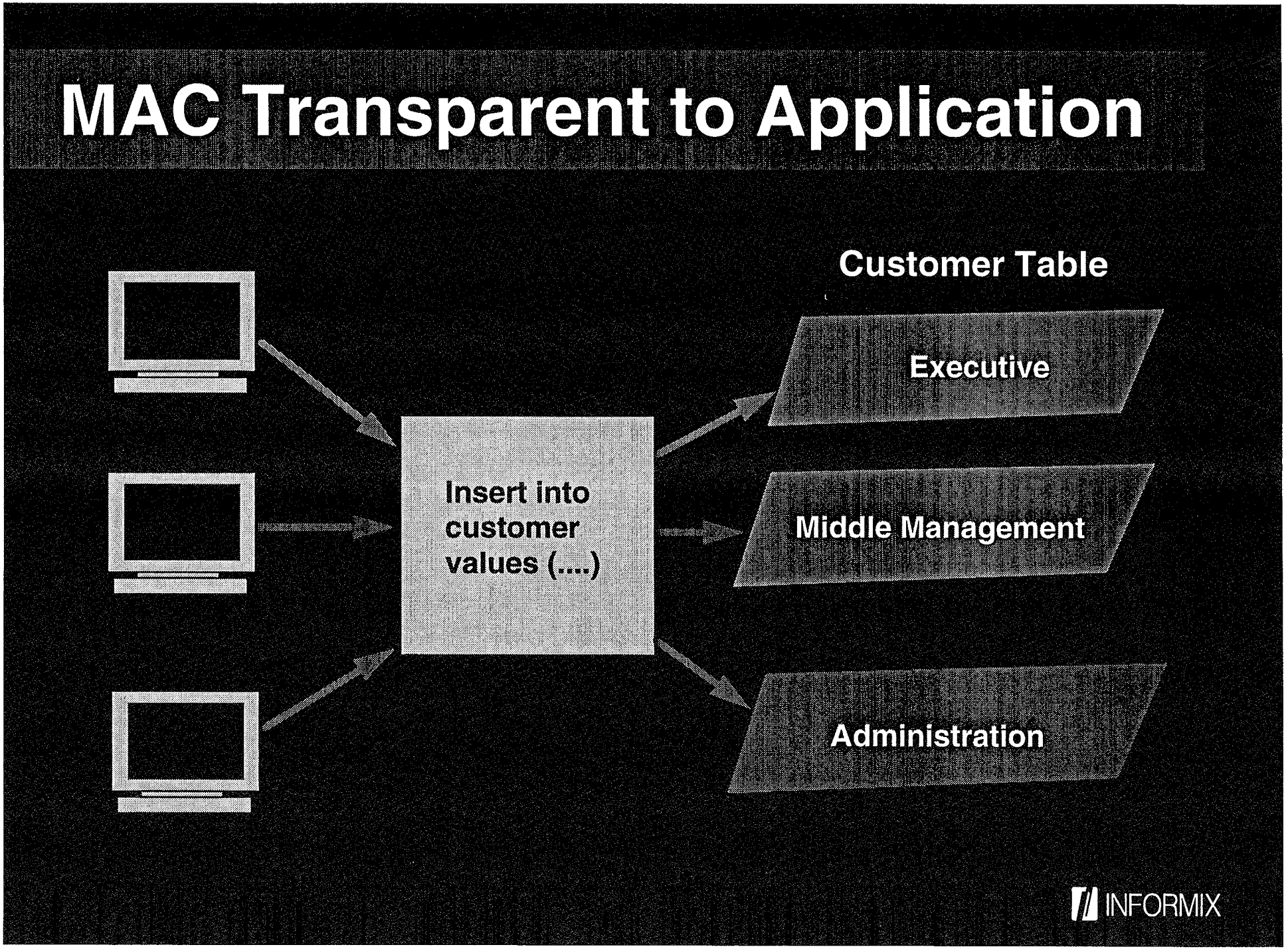
INFORMIX

What is a Secure DBMS?

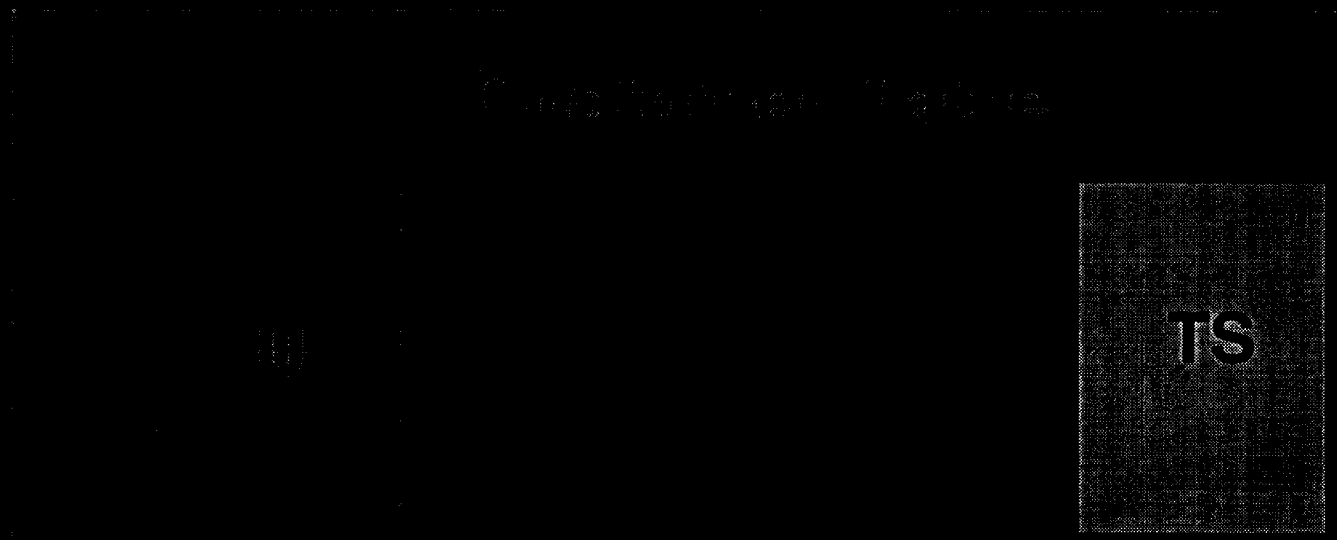
- MAC transparent to the application
- Data is logically and physically separated by level
- No direct label manipulation required

INFORMIX

MAC Transparent to Application



Data Separated by Label



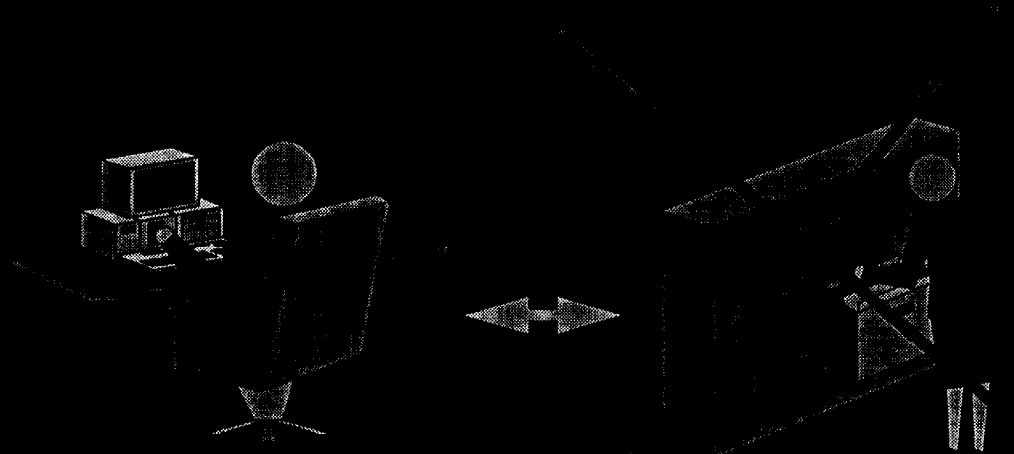
INFORMIX

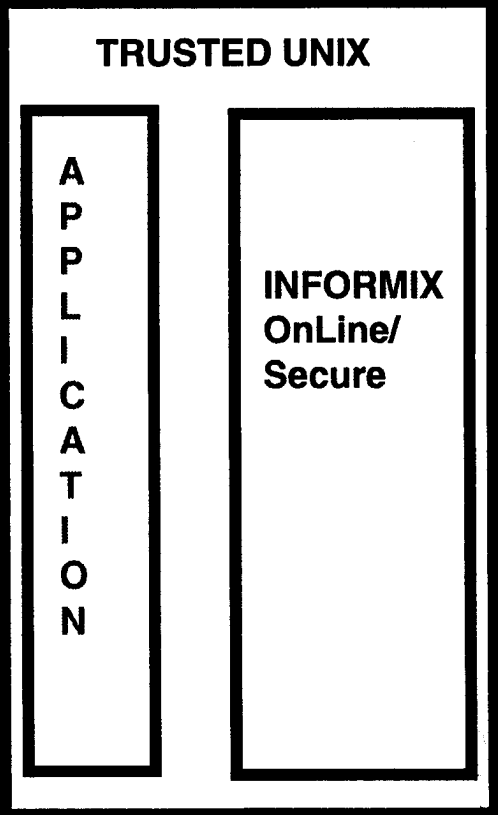
Building Successful Secure Applications

- Define security objectives
- Secure database design — new or exisiting
- Migrate existing applications
- Trusted application design coding and review

INFORMIX

Prevent Unathorized Access





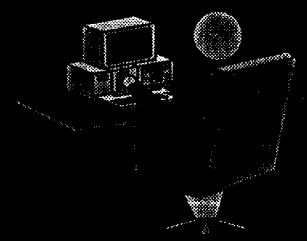
INFORMIX

Prevent Accidental/Deliberate Disclosure

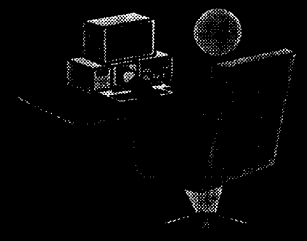
- A secure system will not allow
  - Direct label manipulation
  - Indexes across levels
  - Writing at any level other than current session
  - Reading data at a higher level
  - No multilevel referential integrity relationships

INFORMIX

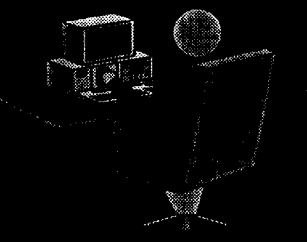
Maintain Data Integrity



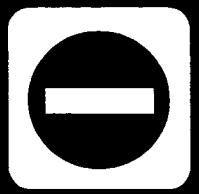
Executive



Middle Management



Adminstration



Corporate Database >= Middle Managers

INFORMIX

Data Migration

- Reclassification of existing data
  - Two methods
    - Load and unload ASCII - high to low
    - Use a trusted application to manage reclassification of data
- This should only be run by the DBSSO

INFORMIX

Secure Database Design

- Design to restrict access
  - Classified databases
  - Classified tables within databases

INFORMIX

Secure Database Design

- Design to ensure correct access
  - Data needed by different clearances of users must be accessible quickly

INFORMIX

Secure Database Design

- Design for performance
  - Multiple levels of data will slow performance when looking down from high levels
  - Heavy auditing can slow performance

INFORMIX

Secure Database Design

- Avoid rows with keys pointing to multiple security levels
- Tables should be designed to reflect security level of users updating the table
- Unique keys should be unique at one level only
- Consider enforcing integrity with check restraints
- Exercise caution when using system generated sequential integers for unique keys in multilevel tables

INFORMIX

Secure Database Design

- Final step of design is determining auditing strategies
  - Per database
  - Per table
  - Per user access to specific data
- Database audit events can be set
  - Default
  - Required
  - Per user

INFORMIX

Building the Secure Applications

- Existing applications
  - Recompile with OnLine/Secure libraries
  - Execute the programs

INFORMIX

Building the Secure Applications

- New applications
  - Design
  - Code
  - Test
  - Security code review
  - Security testing
  - Security officer handles general release

INFORMIX

Building the Secure Applications

- Don't worry about auditing
  - Done by the database and system
  - Controlled by DBSSO and system administrator

INFORMIX

Building the Secure Applications

- Don't worry about user authentication
  - Login, password and clearance retrieved by OS at login
  - Database receives this information from the OS
    - This is TCB (trusted) code for both the OS and OnLine/Secure

INFORMIX

Building the Secure Applications

- All writes automatically done at user's clearance level
  - Inserts
  - Updates
  - Deletes

INFORMIX

Building the Secure Applications

- All reads (selects) automatically retrieve data at or below user's clearance level
  - No "read-up" allowed!
    - Database won't permit this even if program attempts it
  - No lable filters required in SELECT statement

INFORMIX

Building the Secure Applications

- All writes automatically done at user's clearance level (CAVEAT)
  - High clearance user cannot update/delete a lower level row
  - High clearance user logged in at lower level cannot insert/update/delete a higher level row

Building the Secure Applications

- All reads (SELECTS) automatically retrieve data at or below user's clearance level (CAVEAT)
  - Unfiltered SELECTS from a higher level may retrieve more data than expected
    - Label filters/stored procedures can help limited high level queries
    - Environment variable SETSINGLELEVEL can restrict all DML to current level

Be Aware Of:

- Set user ID (suid, guid) functions not allowed
  or restricted to privileged users
  - This is an OS issue
    - Controlled by system administrator/security officer as to who can do this

INFORMIX

Be Aware Of:

- SET SESSION LEVEL can be used to write at different clearances
  - Restricted to privileged users
  - Should not be common in applications

Be Aware Of:

- Singleton SELECTS may return more than one row when run from a high clearance
  - Value look ups may have to be filtered to only look at one level

INFORMIX

Be Aware Of:

- Disk storage requirements
  - Auditing can require a lot of space
  - Many levels of clearance can require a lot of disk space

INFORMIX

Be Aware Of:

- Performance requirements
  - Data stored at many levels will take longer to see from high clearances with an unqualified SELECT

INFORMIX

Building Secure Applications

- Requires a commitment to security from users, developers, management
- Does not have to be difficult
- Does not require good design on the front end
- Fielded applications must be monitored

INFORMIX

DBSim: A Tool for Predicting Database Performance

Mike Lefler and Mark Stokrp
PRC Inc.
1500 PRC Drive
MS: 5S2A
McLean, VA 22102
(703) 556-1863 and (703) 556-1655

1. BACKGROUND

Databases, and particularly relational databases, continue to play an important role in new Department of Defense (DoD) system development efforts. However, despite recent advancements in database and CASE technology, there are currently no accurate tools available to the database designer for reliably predicting performance prior to implementation. In practice, performance issues are typically addressed during acceptance testing after the system is installed and the database is loaded. Therefore, it is not uncommon for the database to initially perform worse than the database it is replacing. DBSim is a tool which applies simulation to address this problem.

The DoD has long recognized the need for developing and applying automated tools for performance evaluation. This is evidenced most recently in the DoD's Defense Technology Plan, a compilation of 19 individual plans that collectively describe the total DoD Science and Technology effort. Plan 8, Computing and Software, includes as one of its goals providing cost-effective tools for developing, managing, and utilizing the high quality software products and systems needed to enable modern DoD strategic and tactical capabilities; and establishes a 1997 milestone for technology for measuring distributed, multi-processor system performance characteristics. The DBSim project is thus a timely, direct response to this key element in the DoD Defense Technology Plan.

2. REQUIREMENTS AND OBJECTIVES

The Database Performance Simulator project has produced a prototype tool that will permit knowledgeable users to model transaction execution prior to coding or system installation and identify the RDBMS strategies and configuration options which optimize performance. More specifically, DBSim:

- Provides a mechanism for describing and simulating a set of transactions against the database.

- Provides a mechanism for describing the size of tables, their index structure, and their allocation to disks.
- Is tunable for different RDBMSs.
- Is tunable for different for CPU speeds, number of CPUs, number of disks and disk latency.
- Allows database designers and engineers to explore "what if" questions and more properly size systems.

DBSim was designed to be easy to interact with, to be parameter driven, and to require minimal expertise to use. It is currently being validated against actual performance measurements of commercial-off-the-shelf (COTS) relational database products in a Sun/Unix environment.

The benefit of DBSim will be that database designers and system engineers will be able to plan and execute the best possible information system for customer environments while mitigating technical and performance risks.

3. HARDWARE AND SOFTWARE ENVIRONMENT

DBSim is programmed in PC SIMSCRIPT II.5, Release 1.8, from CACI Incorporated. This tool was selected for the following reasons:

- It is a general-purpose, rather than an architecture dependent, simulation tool.
- It is a proven, commercially-available product.
- It runs in a PC/MS Windows environment and offers a graphical user interface. Because of economic, transportability, and ease-of-use considerations, a PC-based tool was considered preferable to minicomputer- or mainframe-based tools.

The PC on which this Model was developed included the following features:

- 33 megahertz 80486 microprocessor
- 16 megabytes (MB) of memory
- 320 MB hard disk drive
- 1.44 MB floppy disk drive.

4. DESIGN APPROACH

The major factor which influenced the design of the DBSim prototype was the need to implement a generic tool which can accommodate the range of popular relational DBMSs and accurately simulate their performance. In general, DBSim allows the database designer to:

- Analyze the effects of the size of database tables
- Analyze the effects of the size of database cache (buffer pool)
- Analyze the effects of the index structures
- Analyze the effects of the transaction frequencies
- Analyze the effects of hardware characteristics

By making DBSim modular and parameter-driven, different scenarios of hardware configurations and transaction workloads can be represented by modifying model input parameters.

To ensure modular development, DBSim was designed to exploit the highly layered architecture of the leading relational database management systems (RDBMSs). As depicted in Figure 1, this architecture consists of five layers: User Session, SQL Compilation, Query Execution, Buffer Pool, and Disk Driver. Each layer is implemented as a submodel which calculates execution times through two means: by examining information which is visible only to that layer, and/or by executing a sequence of one or more procedure calls to the next lower layer. The execution times are then propagated upwards.

The *User Session Submodel* emulates the user operational load. It submits model transactions to the system and records the "execution time" returned by the model for later analysis. This model has parameters which allow the modeler to adjust the number of simulated users and the average "think" time between transaction requests. Transactions are chosen stochastically to satisfy transaction ratios specified in advance by the modeler.

The *SQL Compilation Submodel* represents the SQL parser and optimizer components of the RDBMS kernel, mapping each received transaction into a sequence of internal RDBMS operations, e.g., full table scan, indexed scan, temporary table sort. This submodel "processes" the transaction by passing each operation in turn down to the Query Execution Submodel for "execution." The execution times returned by the Query Execution Submodel are saved in a statistical file for later review, summed, and passed back to the User Session Submodel.

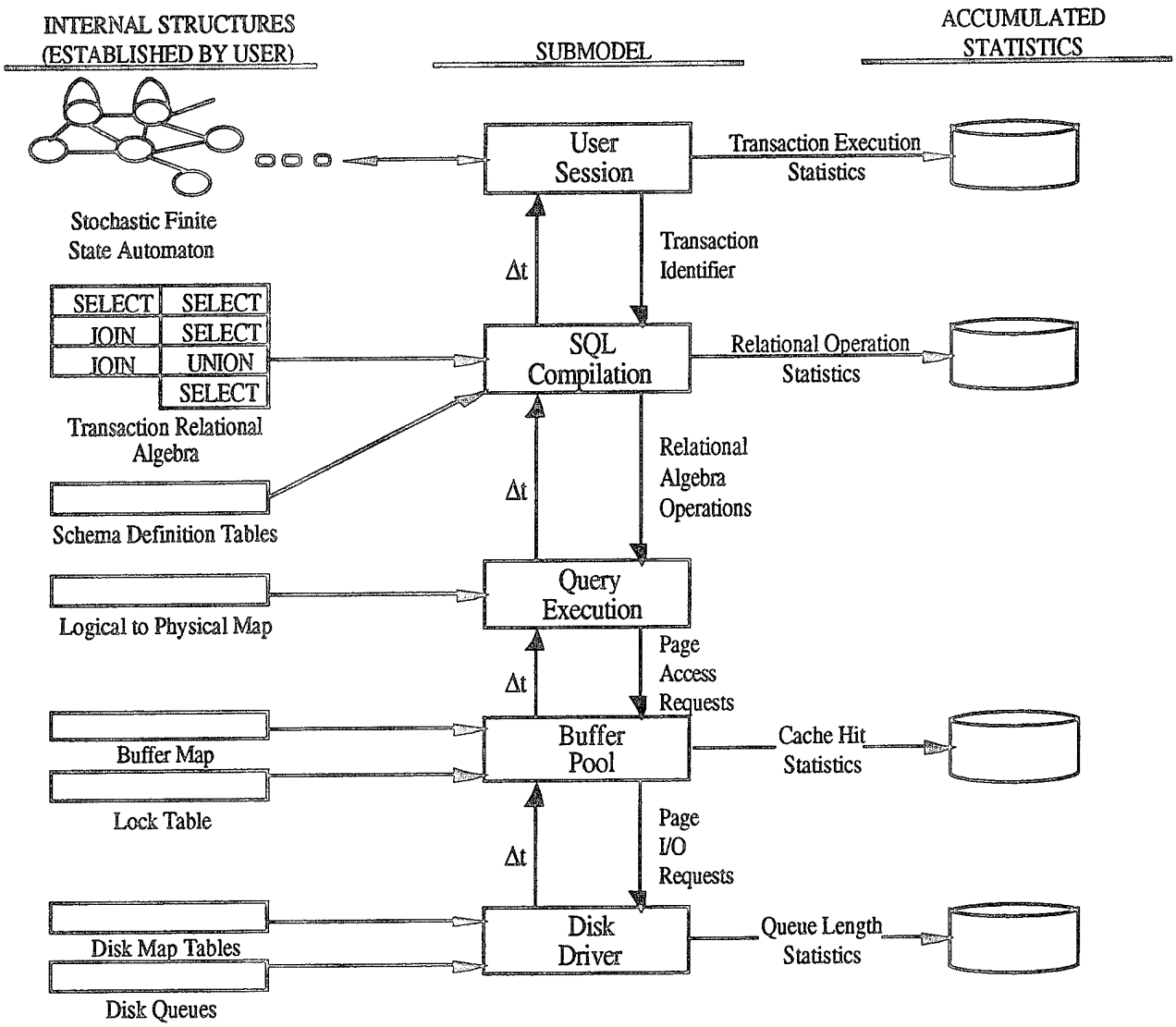


Figure 1: DBSim Layered Architecture

The *Query Execution Submodel* is the heart of DBSim. It compares each primitive internal operation it receives to the physical schema, then generates a sequence of page requests which are passed to the Buffer Pool Submodel.

The *Buffer Pool Submodel* emulates the DBMS's disk cache manager. It receives page access requests in either shared lock (read), or exclusive lock (update) mode. It also emulates the locking strategy and LRU replacement strategy of the disk cache manager and performs deadlock detection.

The *Disk Driver Submodel* receives page I/O requests and uses its internal map to translate page identifiers to disk addresses. It emulates the disk I/O to service the page request.

5. MODEL STRUCTURE

A SIMSCRIPT model such as DBSim consists of three primary components:

(1) A preamble which gives a description of each modeling element and variable,

(2) A main program from which model execution begins, and

(3) A set of events, processes, and routines which describe the occurrences of interest in the system and the actions which take place.

In SIMSCRIPT's terms, an event is an occurrence of interest which stimulates some form of processing in the model, such as the arrival of transactions to a DBMS. A process represents an object of interest and the sequence of actions it experiences throughout its life in the model. A routine describes other actions which take place in the model. The remainder of this section describes key aspects of the model's structure.

Model Resources

In SIMSCRIPT's terms, a resource is an element used to model an object which is required by processes and routines. In DBSim, the resources correspond to the hardware configuration of the system being modeled. If a resource is not available when requested, the request is placed in a queue and made to wait until the resource becomes available.

Model Control

DBSim was designed with a Control process whose purpose is to correctly direct the flow of each transaction through the layers of the emulated RDBMS and keep track of the time spent waiting for and utilizing each resource. In DBSim, the Control process directs the flow of each transaction through the RDBMS in accordance with the processing flow illustrated in Figure 1.

Model Statistics Collection

SIMSCRIPT provides a number of built-in facilities for collecting performance statistics of interest during the simulation. Statistics such as averages and maximums on the queues and the utilization of resources are most commonly of interest. In DBSim, these built-in facilities were used to automatically track the average and maximum queue lengths and utilization of each resource. However, there were other performance statistics of interest to us. Consequently the built-in facilities of SIMSCRIPT were augmented to collect these additional statistics. As the Control process directs each transaction through the DBSim model, it also updates the appropriate data items within a special data structure, which, along with the

statistics automatically tracked by SIMSCRIPT, are then used to generate the output reports.

Model Output

During simulation runs, DBSim maintains an on-screen graphical display depicting the performance of the system and tracking key performance measures. In addition, DBSim generates three output reports. As shown in Figure 2, the first report summarizes system throughput by transaction class. For each transaction class, the report shows the number of transactions executed by type, both total transactions and per-unit-time averages. It also records the buffer pool hit ratio and total disk I/O commands.

PRC DATABASE PERFORMANCE SIMULATOR

DATABASE THROUGHPUT SUMMARY STATISTICS

TRANS TYPE:	TOTALS RECEIVED	TOTALS PROCESSED	AVERAGES PER HOUR	AVERAGES PER SEC
1	2399	2399	299.87	.08
2	9599	9599	1199.88	.33
3	1599	1599	199.88	.06
4	799	799	99.88	.03
5	3199	3199	399.87	.11
TOTALS:	17595	17595	2199.37	.61

DATABASE BUFFER HIT RATIO WAS 98.3 %

TOTAL NUMBER OF DISK READS WAS 15865

TOTAL NUMBER OF DISK WRITES WAS 3908

AVG NUMBER OF CONCURRENT DATABASE THREADS WAS .8
MAX NUMBER OF CONCURRENT DATABASE THREADS WAS 7.0

Figure 2: Throughput Summary Statistics Report

The second DBSim output report summarizes state probability statistics by transaction class. As shown in Figure 3, this report is read vertically for each transaction class. Of the total time spent within the system by each transaction class, the report shows the proportion of that time spent in each of the 10 states listed

along the left column of the report. Because of rounding, the state probabilities in each column may not sum exactly to 1.00.

PRC DATABASE PERFORMANCE SIMULATOR

STATE PROBABILITIES BY TRANSACTION TYPE

STATE:	1	2	3	4	5	6
WAITING USER SESSION	.00	.00	.00	.00	.00	
USER SESSION	.01	.01	.01	.01	.01	
WAITING SQL COMPILATION	.08	.03	.01	.01	.05	
SQL COMPILATION	.24	.16	.20	.13	.10	
WAITING QUERY EXECUTION	.28	.04	.20	.27	.09	
QUERY EXECUTION	.24	.12	.30	.34	.09	
WAITING BUFFER POOL	.00	.00	.00	.00	.00	
BUFFER POOL	.00	.00	.00	.02	.00	
WAITING DISK DRIVER	.04	.02	.04	.06	.19	
DISK DRIVER	.10	.62	.24	.16	.47	

(NOTE: DUE TO ROUNDING, THE PROBABILITIES IN EACH COLUMN MAY NOT SUM EXACTLY TO 1.00)

Figure 3: State Probabilities Report

Examining the State Probabilities Report allows an assessment to be made as to whether the system has been sized adequately to handle expected workloads and to identify potential system bottleneck areas. Areas in which the transactions spend significant proportions of time are typically of prime interest to database designers and systems engineers.

As a companion to the State Probabilities Report, the Resource Utilization and Queuing Statistics Report depicted in Figure 4 shows the capacity units available, average utilization, average queue size, and maximum queue size for each resource over the period simulated. Examining these utilization and queuing statistics allows a systems engineer to determine whether the system has been sized adequately to handle expected workloads, and to identify potential system bottleneck areas.

PRC DATABASE PERFORMANCE SIMULATOR

RESOURCE UTILIZATION AND QUEUING STATISTICS

RESOURCE:	UNITS	MEAN UTIL	MEAN QUEUE	MAX QUEUE
DBMS.CPU (1)	1	.097	.068248	2
DBMS.CPU (2)	1	.148	.039975	1
DBMS.DISK (1)	1	.320	.049424	2
DBMS.DISK (2)	1	.073	.005150	2
DBMS.DISK (3)	1	.091	.005109	3

THE MEAN NUMBER OF BUFFER POOL PAGES IN USE WAS 300.0 OUT OF 300 (100.0

THERE WERE A TOTAL OF 0 FORCED DIRTY WRITES TAKING A TOTAL OF 0. SECONDS

THE TOTAL NUMBER OF BUFFER POOL DEADLOCKS ENCOUNTERED WAS 0

S I M U L A T I O N C O M P L E T E D
Ending Date and Time: 10/17/1994 at 08:40:40

Figure 4: Resource Utilization and Queuing Statistics Report

Biographies:

Mike Lefler, PRC Technology Center, PRC Inc., 1500 PRC Drive, McLean, VA 22102, 703-556-1863 (lefler_mike.@prc.com). As a Senior Technical Fellow Mike Lefler serves as an in-house consultant on database technology for PRC, a major systems integrator. In this capacity he supports a wide spectrum of PRC programs. In addition he edits a column devoted to DBMS technology for PRC's *Technology Transfer* newsletter, and he has presented seminars on relational database technology, parallel database technology, information engineering methodology, and distributed databases as part of PRC's internal training program.

Mark Stokrp, PRC Technology Center, PRC Inc., 1500 PRC Drive, McLean, VA 22102, 703-556-1655 (stokrp_mark.@prc.com). As a Senior Technical Fellow Mark Stokrp provides in-house consulting and technical support services across a wide spectrum of programs in the areas of modeling and simulation, computer performance evaluation, system sizing, capacity planning, and performance measurement for PRC, a major systems integrator. Recently he chaired PRC's 1995 Spring Technical Seminar on Advanced Simulation for Acquisition, Operations, and Training.

IMPROVING SECURITY IN MULTI-LEVEL DATABASE MANAGEMENT SYSTEMS THROUGH THE USE OF QUERY MODIFICATION

Michael Lawrence Martin, Ph.D., DISA/JIEO/Center for Software

ABSTRACT

Current security models are overly complex, inefficient and make it difficult to implement security constraints. This paper demonstrates a new and more efficient mechanism drawn from distributed database technology for secure query processing in multi-level databases implemented as kernelized architectures. The paper develops a query modification process at the fragment level for these Database Management Systems (DBMS). Then it adapts and applies techniques of distributed databases for transforming global queries into query fragments. The resultant list of query fragments can then be parsed to remove any query fragments which would violate security constraints. The remaining list of query fragments can then be safely processed against single-level database fragments. The resultant safe query responses are then combined to form a secure response to the user's query.

Most of the research in computer security through the early eighties concentrated on secure operating systems. Because of the advent of DBMS as the primary form in which to maintain information, extending multi-level security to databases became apparent. The paper reviews three models that have attempted to extend security to multi-level databases: the SeaView, Commutative Filter, and Query Modification Models. The paper concludes that the existing models rely on inefficient procedures to enforce security rules. For example, the commutative filter approach allows all queries to go to the database -- even those which clearly violate security rules -- and then parses the answer to eliminate data elements which should not be disclosed [Denn85]. In some cases, all records may be filtered out after being returned from the database. The query modification approach advocated by Keefe [KeeAL89] for multi-level databases implemented as kernelized architectures searches security constraint tables for all relevant security rules concerning any of the data elements prior to searching the database. Since the single-level fragments are uniformly classified and distributed to these levels in accordance with these rules, this approach is also overly complex.

Each of these models is analogous to having a distributed database based on security level. However, none of these models has used the inherent advantages of distributed query processing techniques to access or to provide security to the single-level fragments. A multi-level database consisting of relations which are fragmented based on security level is logically the same as a distributed database. The paper uses the equivalence expressions developed by query optimization research to modify queries for security purposes.

1. INTRODUCTION

Most of the research in computer security through the early eighties concentrated on secure operating systems. Because of the advent of DBMS as the primary form in which to maintain information, extending multi-level security to databases became apparent. In 1983 the Air Force sponsored a conference of the leading researchers in the database and security field to define a set of goals and approaches to multi-level secure DBMSs.

The resultant Air Force Study [CMDM83] noted the vulnerability of databases to inference and aggregation problems. Inference is the ability of a user to deduce the value of data hidden because of security from other information. Aggregation is concerned with groupings of data which become sensitive while the individual data are not. The typical example of an aggregation problem is an agency's phone book which is classified while the individual phone numbers of employees are not. The Air Force Study recognized that databases operated on smaller units of data (lower granularity), such as tuples, attributes, and elements, than did operating systems. The Trusted Database Management System Interpretation (TDI) extended the trusted computer system evaluation criteria to database systems [NCSC90]. The Air Force Study also recommended that several

near term architectures for Multi-Level Secure (MLS) DBMSs be investigated and that a security model based on the view mechanism be investigated.

The SeaView model was developed as a prototype of a MLS DBMS based on the view mechanism. In the SeaView model, data elements are classified at the element-level and stored as single-level fragments. The single-level fragments are created from multi-level relations which are partitioned based on security level. Each of the single-level fragments contains the primary key of its parent relation. When a user requests information, a view is created consisting of those single-level fragments that the user is authorized to access.

Later studies corrected errors and improved on the SeaView model's algorithms for processing these single-level fragments [JajSan90a/b] or proposed alternative algorithms to accomplish the same purpose [JanSan91]. Another approach, the Commutative Filter Model, advocates replacing the security kernel that enforces mandatory access control with a trusted filter which would intercept returning query answers from the database and remove all data the user is not entitled to see [Denn85]. A still other approach, the Query Modification Model, advocates replacing the security kernel with a trusted query modification process which would intercept queries before they are processed by the database and modify them so that the resulting query would not result in any unauthorized disclosure [KeeAL89]. The three models, SeaView, Commutative Filter, and Query Modification, are discussed in detail in Section 2 and are referenced throughout this paper.

This paper addresses two problems identified in current models to ensure security in multi-level database systems:

First, the models discussed above rely on inefficient procedures to enforce security rules. For example, the commutative filter approach allows all queries to go to the database -- even those which clearly violate security rules -- and then parses the answer to eliminate data elements which should not be disclosed [Denn85]. In some cases, all records may be filtered out after being returned from the database. The query modification approach advocated by Keefe [KeeAL89] for multi-level databases implemented as kernelized architectures searches security constraint tables for all relevant security rules concerning any of the data elements prior to searching the database. Since the single-level fragments are uniformly classified and distributed to these levels in accordance with these rules, this approach is also overly complex.

Each of these models is analogous to having a distributed database based on security level. However, none of these models has used the inherent advantages of distributed query processing techniques to access or to provide security to the single-level fragments. This paper determines the applicability of these techniques and, as discussed below, demonstrates their utility.

Second, these models exclude information that could be provided to the user without violating security constraints. In order to prevent certain kinds of inference, the models exclude entire tuples when some of the individual attributes could be provided to the user without any compromise of security.

The remainder of the paper is organized as follows:

- Section 2 presents a review of related research which has influenced the paper research.
- Section 3 demonstrates the conversion of user queries to safe queries in multi-level databases implemented as kernelized architectures.
- Section 4 summarizes the contributions of this paper and presents areas requiring further research.

2. RELATED RESEARCH

Research related to the topics in this paper falls into two broad categories: multi-level security research

and query processing research.

2.1 Research on Multi-level Security

By the early 1980s the desirability of extending multi-level security to DBMSs was apparent. The Multi-level Data Management Security Study [CMDM83] reports the results of a workshop on MLS databases by some of the leading researchers in the field. The study recognized that implementing MLS in databases presented a more complex challenge since databases operate on a lower level of granularity than do operating systems.

Security models have been proposed for all levels of granularity. The I.P. Sharp Model [Groh76] proposed classification at the relational-level. The Hinke-Schaefer Model [HinSch75] proposed classification at the attribute-level. The TRW Model [Grav86] proposed classification at the tuple-level. The SeaView Model [DenAL86, DenAL87a, DenAL88, LunAL88] proposes classification at the element-level of granularity.

Of all of these models the SeaView model has had the most influence on this paper. The SeaView model is formulated in two layers, an inner layer which provides Mandatory Access Control (MAC) and an outer Trusted Computing Base (TCB) which provides Discretionary Access Control (DAC). The inner layer satisfies the Orange Book requirements for MAC. It provides the access control mitigating the access requests of the users of the data (comparing the clearance of the users with the level of the requested data). This inner layer corresponds to a security kernel or reference monitor that meets the criteria for the highest level of trust described in the Orange Book (A1). The outer layer or TCB adds the DAC and contains the components of a multi-level-secure relational database system. The proponents of SeaView argue that it meets the TDI requirements for certification as a multi-level secure DBMS because of its layered approach.

The SeaView model has two key features. First, it extends the basic relational model by including classification attributes for each elemental data item. Thus, each elemental data item is now a couple {Data Element, Classification}. This allows rules for labeling new data to be expressed as integrity constraints on the classification attributes, and retrievals can select on values for classification attributes as well as data attributes. The SeaView model also extends the definition of entity integrity to multi-level relations. In particular, the model requires the uniform classification of primary keys.

Second, the model decomposes all multi-level relations into single-level relations which are stored in single-level fragments. The original SeaView decomposition scheme had several problems which were documented by Jajodia and Sandhu [JajSan90a/b] The identified problems included: repeated joins, spurious tuples, incompleteness, and left outer joins. The first three of these problems were eliminated by modifying the decomposition and recovery algorithms [JajSan90a/b]. These modified algorithms are used in the SeaView discussions in this paper. The left outer join problem is eliminated by using the novel decomposition algorithm proposed in [JajSan91]. The novel decomposition algorithm replaces the left outer join operations with union operations which the authors claim are more efficient than outer joins.

Aside from SeaView itself, two other models have had significant influence on this paper: the Commutative Filter Model [Denn85] and the Query Modification Model [KeeAL89]. Both of the models replace the underlying secure operating system protected by a security kernel with the equivalent for the DBMS system. This offers an advantage since DBMSs were originally designed to bypass many of the underlaying features of operating systems because they were too slow or did not offer solutions to DBMS problems. Here the idea is to isolate those features of the DBMS that are related to security and place them in a trusted component which will provide security.

In the commutative filter approach, queries posed by the users are augmented with classification and checksum fields and passed to the DBMS. The returning answers are examined by a trusted commutative filter which excludes any data which the user is not entitled to see. The commutative filter does this through the use of what Denning calls the authorized view equivalence scheme. The query response returned to the user is made equal to the maximal authorized view (for a given user). Denning developed algorithms for providing a secure

query response (qsec) for three levels of granularity: tuple-level, attribute-level , and element-level .

Unfortunately, the Denning approach is more restrictive than required. For example, if a user query includes any attribute that the user is not entitled to see, the entire query is rejected (under attribute-level labeling). This approach requires users to know the names of each and every attribute they are authorized to see in order to formulate a query which will be answered. This approach also eliminates implicit query forms such as: Select * from Emp (where * means all attributes).

The restrictions of the commutative filter approach are more severe at the element-level of labeling. The query response is screened and entire tuples are suppressed if any element value would be null due to a request for unauthorized data. The filtering process often eliminates considerable information in order to eliminate inferences which otherwise might be made. This paper will demonstrate, however, that the filtering approach restricts more information than is necessary to preserve security.

The query modification approach offers a potential solution to this problem. In contrast to the filtering approach, where security criteria are imposed after the response is returned, in the query modification approach [KeeAL89], queries posed by the users are intercepted by a trusted query modification process. The original user query is examined as is the clearance of the user and the classification of the data being queried. Then, the query is determined to be safe (with respect to security) and is passed to the DBMS, or the query is modified to a safe form. The modification entails transforming the query from its original form to a new safe (secure) form.

Query modification as suggested by Keefe requires the development of a very complex, query modification process. The Keefe technique involves searching a security constraints database for all rules involving the query in question. The restrictions in these rules are added to the original query in order to make a safe query. This process takes exponential time with respect to the number of attributes released by the query. In addition, some of the resulting safe queries would have little or no meaning to the user.

2.2 Query Processing

Query optimization is a process of manipulating the original query expressed by a user in order to improve the efficiency of the query (reduce the execution time). The idea of query optimization is to avoid operations which take a long time (such as joins and, especially, cartesian products) and substitute operations which are quicker (such as performing selections and projections before joins). The transformations are performed by substituting expensive operations with their more efficient equivalent expressions.

Query transformation research has also been extensively done in the area of distributed databases. In this case, the global query is transformed into a set of equivalent fragment queries.

The studies of query processing that have had the most influence on this paper are those involving distributed query processing [CerAL83, CerAL82, CerPel83, CerPel84, and ChaChe80]. These papers contain detailed treatments of the process of converting global queries into their horizontal, vertical and mixed fragment equivalents. These query fragments are processed at the appropriate databases. While the process was developed for reasons other than security (for example, economies of scale), the theory behind query processing in distributed databases is useful in ensuring security in multi-level databases. The fragmentation schemes used in SeaView and the other secure database models in effect create horizontal, vertical and mixed subrelations. Thus, the means chosen to ensure security is analogous to the processes chosen for query processing in distributed databases.

This paper builds on this recognition of the similarities in the two processes for the treatment of single-level fragments. It uses the equivalence expressions developed by this query optimization research to modify queries for security purposes.

3. CONVERTING USER QUERIES TO SAFE QUERIES FOR MULTI-LEVEL RELATIONS IMPLEMENTED AS KERNELIZED ARCHITECTURES

In multi-level databases, data elements and users are classified at different security levels. In such an environment, it is necessary to prevent users at a given clearance level from accessing data at a higher classification. Queries made by subjects can be intercepted and modified so that a subject may retrieve only the information for which the subject has the appropriate clearance.

The query modification process can be viewed as subtracting requests for information above the subject's level from the original query. This process continues until there are no requests for unauthorized information. When the query no longer contains requests for unauthorized information, it may be defined as a *safe query* with respect to the subject's security level. Thus, a query is safe if all data elements involved in the query are dominated by the clearance of the subject. If the user's query is not safe, the techniques of query modification are used to make it safe.

To preserve security in a multi-level database environment, each query must be transformed into a safe or secure form with respect to the subject making the query unless the subject's clearance level is the highest level on the security lattice. If a subject is cleared at the highest level, it has access to all information in the database, and any query posed by the subject is safe with respect to that subject.

In multi-level relations, the attributes have different security levels. Multi-level relations can be decomposed into single-level relations and stored in kernelized architectures [DenAL87a-c, JajSan91, KeeAL89]. This paper uses Jajodia and Sandhu's definition of a multi-level relation for kernelized architectures [JajSan91]. A multi-level relation has a relation scheme which is represented by: $R(A_1,C_1,A_2,C_2,...,A_n,C_n,TC)$. Each data attribute ($A_i$) is paired with a classification attribute (C_i) and, in addition, a classification attribute for each tuple (TC). The domain of C_i is the security lattice from L_i to H_i. The domain of TC is [lub{L_i: i= 1...n},lub{H_i: i= 1...n}] where lub is the least upper bound. Each instance of the relation R has the form $R_c(A_1,C_1,A_2,C_2,...,A_n,C_n,TC)$ with *c* representing each access class in the security lattice. Alternately, the relations can be stored whole in their multi-level form.

Kernelized implementations require decomposition algorithms which break the multi-level relations into a number of single-level relations. Queries against multi-level databases that are implemented with kernelized architectures may be processed in one of two ways.

First, the queries may be processed against a "secure" global view which has been filtered by a "trusted system" or reference monitor as advocated by Denning [Denn85]. Second, the queries may be fragmented according to the decomposition scheme and processed against the single-level relations. The first approach was discussed in section 2. The second approach will be discussed in detail in this section.

The discussion of the fragmentation approach to query modification begins with a precise exposition of what is meant by a safe query in this context:

> ***Definition:*** A multi-level query Q is *safe* with respect to subject S (the subject making the query) logged in at level L(S) if after parsing the query, the resulting relational algebra expression is safe. A relational algebra expression is safe if it only involves relations $R_l,..., R_h$ such that $L(R_i) \leq L(S)\ \forall\ l \leq i \leq h$, where $L(R_i)$ is the classification level of R_i.

In other words, the query is safe if the classifications of all relations involved in the query are dominated by the clearance of the subject. The subject may log in at any security level for which he/she has been cleared. In order to simplify the examples and explanations in this paper it is assumed that the subject's clearance level designated L(S) is the level that the subject logged in at for the transaction or session being discussed.

In this definition, it is assumed that the relations involved in the query are single-level fragments of the original multi-level relations. This definition restricts access to those fragments whose classification level is

dominated by the subject's clearance level. Thus, any selection criterion involving an attribute with a classification level not dominated by the subject's clearance level is automatically eliminated.

This section shows how user queries are converted to safe queries when multi-level relations are implemented using kernelized architectures. The section demonstrates that secure queries can be formed using the inherent features of the decomposition scheme used to create the fragments. The conversion process is accomplished using the following steps [CerPel84, Date90]:

1) The SQL query is converted into relational algebra for purposes of internal representation.

2) The relational algebra is converted into a form that will recover the multi-level relation from the fragments using the union, natural join, and union operations for tuple, attribute, and element level granularity respectively.

3) The appropriate fragments are substituted into the recovery formula (incompatible fragments and fragments not dominated by the log-in level of the user are excluded).

4) The various relational operations are distributed to the appropriate fragments.

The conversion process recovers a multi-level relation from its fragments regardless of the level of granularity of the fragments. Following a brief review of notations and assumptions for the section as a whole, the remaining sections demonstrate recovery of a multi-level relation from its fragments for each labeling granularity: tuple, attribute and element. Each of the sections shows the safe query process for the three basic relational algebra operations: selection, projection, and natural join.

3.1 Notations and Assumptions in Converting Queries for Kernelized Architectures

The formulas presented in this section assume R and W are multi-level relations with attributes $A_1,...,A_n$ for R and $B_1,...,B_m$ for W.

Whatever the labeling granularity, the relations R and W can be partitioned into a set of fragments containing data classified at one level only, using a decomposition algorithm. The specific algorithms used differ depending on the level of granularity used for security labeling. The notation F stands for the formula which is used to decompose the multi-level relations into single-level fragments -- $F(R) \rightarrow R_l, \ldots, R_k, \ldots, R_h$, where l is the lowest and h is the highest classification level in the security lattice, and k is a classification level between l and h.

To combine the fragments into the original multi-level relation or a multi-level subset of the original relation, a recovery algorithm, the inverse of F, is used. The recovery algorithm to combine fragments of R at levels l through h into one multi-level relation R is as follows: $F^{-1}\{R_{l:h}\} \rightarrow F^{-1}\{R_l, \ldots, R_h\}$. The symbol (->) is used as the notation for "converts to." Depending on the labeling granularity, the recovery algorithm uses the union, join, or a combination of union and join operations to combine the single-level fragments into a multi-level relation.

A relational algebra expression on a multi-level relation is transformed into a set of equivalent relational algebra expressions on the fragments. This process of conversion is similar to converting a global query into sub-queries in a distributed database [CerPel84]. In the case of multi-level relations the distribution algorithm used to create the fragments is based on the security levels in the lattice. During the process of converting the global query into sub-queries, the security classification of the original query is appended to each sub-query so that the level of each sub-query can be compared to the level of the fragments. All fragments which are not dominated by the log-in level of the subject, i.e., the level at which the subject logged in, are excluded from the multi-level relation. The result is a safe query against the recovered multi-level relation since it was created from only those fragments dominated by the subject S logged-in at L(S).

3.2 Tuple-Level Granularity

When classification is at the tuple-level, individual attributes/elements do not have to have security labels. In the tuple-level model, each tuple in R and W has an additional attribute, TC, specifying the classification level of the entire tuple. To prevent unauthorized access to data, the multi-level relations R and W are horizontally fragmented using a decomposition algorithm, F, that creates a set of disjoint fragments of each multi-level relation for each classification level i (where $l \leq i \leq h$) in the security lattice. This horizontal fragmentation scheme can be defined by expressing each fragment as a selection operation on the original multi-level relation. The selection criterion specifies all tuples in the multi-level relation which are classified at a particular classification level i. Therefore, multi-level relations R and W are decomposed into their corresponding single-level fragments R_i and W_i as follows:

$$R_i = \sigma_{TC=i}(R) \quad \forall\, l \leq i \leq h$$

The recovery algorithm, the inverse of F, uses the union operation to combine the single-level fragments into the original multi-level relation:

$$R = \cup [R_i] \quad \forall\ l \leq i \leq h$$

When tuple-level labelling is used, each relational algebra expression is transformed into a safe expression with respect to a subject S logged-in at L(S) by replacing each multi-level relation in the expression with the union of those fragments of the multi-level relation which are dominated by L(S).

As an example to demonstrate the above decomposition and recovery schemes, assume that an instance of relation Employee (EMP) with attributes SSN, Name, Salary, and Department_Number (DN) [where SSN is the primary key and TC is the classification of the tuple] exists as follows:

Employee:

SSN	Name	Salary	DN	TC
001	Able	20000	C100	U
002	Bond	45000	C200	C
003	Coe	70000	C300	S
004	Drake	135000	C400	TS

The decomposed relation is as follows:

D_U:

SSN	Name	Salary	DN
001	Able	20000	C100

D_C:

SSN	Name	Salary	DN
002	Bond	45000	C200

D_S:

SSN	Name	Salary	DN
003	Coe	70000	C300

D_{TS}:

SSN	Name	Salary	DN
004	Drake	135000	C400

To recover the multi-level relation R for a user logged-in at the Secret level:

$$R_S = D_U \cup D_C \cup D_S$$

(Note: during recovery the tuple classification attribute TC is set equal to the classification level of the source subrelation, e.g., tuples from subrelation D_U have the tuple classification attribute set to U -- Level(TC) = Level(D_i))

R_S:

SSN	Name	Salary	DN	TC
001	Able	20000	C100	U
002	Bond	45000	C200	C
003	Coe	70000	C300	S

In the examples in this section, the Employee relation defined above will serve as the first relation (R relation). The Department relation (or the W relation) will serve as the second relation. The Department relation has attributes DN, Dname, Project, and Manager (where DN is the primary key). The tuple classification attribute for the Department relations will be designated TE.

Department:

DN	Dname	Project	Manager	TE
C100	Computer	Software	Smith	U
C200	Crypto	Sneaky	Jones	C
C300	Foreign	Spying	Bond	S
C400	Nuclear	Weapon	Green	TS

The Department relation is decomposed and recovered in the same manner as the Employee relation.

The following sections show query optimization for the three relational algebra operations: Selection, Projection, and Natural Join . For additional information, see [Mart94].

3.2.1 Selection. The following formula states that performing a secure selection operation (σ_F) on a multi-level relation R_j is equivalent (->) to performing a selection operation over the union (U) of all of the decomposed single-level subrelations (D_i) which are dominated by the security level of the user [L(S)], which in turn is equivalent to taking the union of each subrelations after the selection has been made on the subrelation.

$\sigma_F R_j$ -> σ_F {U D_i} for $l \leq i \leq L(S)$; where j is max(i)

which is equivalent to

U {$\sigma_F D_i$} for $l \leq i \leq L(S)$

by the distributivity of the selection over the union operation.

The subscript i refers to the security level of each subrelation D_i. The subscript j represents the node in the security lattice equivalent to the log-in level of the subject [L(S)] which dominates all of the i. The resulting relation R_j is secure since all of the subrelations used to form it were dominated by L(S).

3.2.2 Projection. The function F is usually expressed as a simple list of the attributes to be included in the response, i.e., {A_j, ... , A_m}. The following expressions are all equivalent. The same format is used as was detailed in the selection section.

π {A_j, ... , A_m} R -> π {A_j, ... , A_m} [U {R_i}]

where $j \geq 1$, $m \leq n$, and $l \leq i \leq L(S)$

Equivalently, distributing the projection over the union

π {A_j,...,A_m} R -> U {π (A_j,...,A_m) R_i}.

The above statement translates as: To perform a secure projection operation (π) on a set of attributes {A_j, ... , A_m} over a multi-level relation R is equivalent to the projection of those attributes over the union (U) of the subrelations making up R which are dominated by the security level of the user L(S) which is equivalent to taking the union after projecting the desired attributes from each subrelation dominated by L(S).

3.2.3 Natural Join. Either of two equivalent formulas may be used to arrive at the secure natural join ($\bowtie$) of two multi-level relations, R and W. The statement below translates as: To perform a secure natural join operation ($\bowtie$) on two multi-level relations R & W is equivalent to taking the natural join after the union of all subrelations R dominated by L(S) and the union of all subrelations W dominated by L(S) which is equivalent to taking the union of the natural joins of all possible combinations of subrelations.

R $\bowtie$ W -> [U {R_i}] $\bowtie$ [U {W_i}], where $l \leq i \leq L(S)$

Equivalently, distributing the natural join over the union

R $\bowtie$ W -> U [R_i $\bowtie$ W_i], where $l \leq i \leq L(S)$

3.2.4. Secure Query Process Against Tuple-Level Granularity

Figure 1 shows the secure query process for queries made against tuple-level granularity for a user logged-in at the secret level. This query combines the Selection, Projection, and Natural Join rules given above.

Figure 2 shows the secure query response to the query in Figure 1.

Figure 3 shows the secure query process in query tree form. The query tree form illustrates the pruning process by showing the edges removed to all fragements whose security level is not dominated by the level of the subject.

SELECT Name, Salary, DN, Project
FROM Employee, Department
Where Salary > 30,000

$Q_S = \sigma_{Salary > 30,000}\ \pi_{Name, Salary, DN, Project}$ Employee $\bowtie$ Department

-> $\sigma_{Salary > 30,000}\ \pi_{Name, Salary, DN}\{\cup \text{Employee}_i\} \bowtie \pi_{DN,Project}\ \{\cup \text{Department}_i\}$

which equals: $\sigma_{Salary > 30,000}\ \pi_{Name, Salary, DN}\ (\text{Employee}_U \cup \text{Employee}_C \cup \text{Employee}_S) \bowtie \pi_{DN,Project}(\text{Department}_U \cup \text{Department}_C \cup \text{Department}_S)$

which is equivalent to: $\sigma_{Salary > 30,000}\ \pi_{Name, Salary, DN}\ \text{Employee}_U \cup \sigma_{Salary > 30,000}\ \pi_{Name, Salary, DN}\ \text{Employee}_C \cup \sigma_{Salary > 30,000}\ \pi_{Name, Salary, DN}\ \text{Employee}_S \bowtie \pi_{DN,Project}\ \text{Department}_U \cup \pi_{DN,Project}\ \text{Department}_C \cup \pi_{DN,Project}\ \text{Department}_S$

Figure 1
Secure Query Process Against Tuple-Level Granularity

$\sigma_{Salary > 30,000}\ \pi_{Name, Salary, DN, Project}$ Employee $\bowtie$ Department

Name	Salary	R.DN	Project	TF
Bond	45000	C200	Sneaky	C
Coe	70000	C300	Spying	S

Figure 2
Secure Query Response for Tuple-Level Granularity

3.3 Attribute-Level Granularity

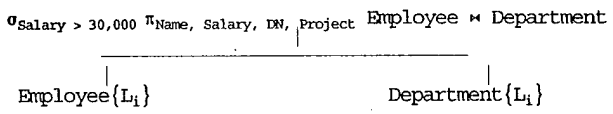
In the attribute-level model, an additional attribute C_i, specifying the classification level of the attribute, is associated with each attribute A_i and B_i in relations R and W, respectively. To prevent unauthorized access to data, R and W are vertically fragmented using a decomposition algorithm F that creates a fragment of each relation for each classification level i (where $l \le i \le h$) in the security lattice. This vertical fragmentation can be defined by expressing each fragment as a projection of all attributes of the multi-level relation classified at security level i. In order to facilitate the reconstruction of the original multi-level relation using the recovery algorithm, each single-level fragment must include the primary key of the relation in addition to the attributes classified at a particular level. In this paper the primary keys for R and W are designated as A_1 and B_1, respectively. The classification attributes for the primary keys are designated as $C(A_1)$ and $C(B_1)$. The fragmentation is correct if each attribute except the primary key is mapped into exactly one attribute of exactly one fragment.

Therefore, relations R and W are decomposed into their corresponding single-level fragments as follows:

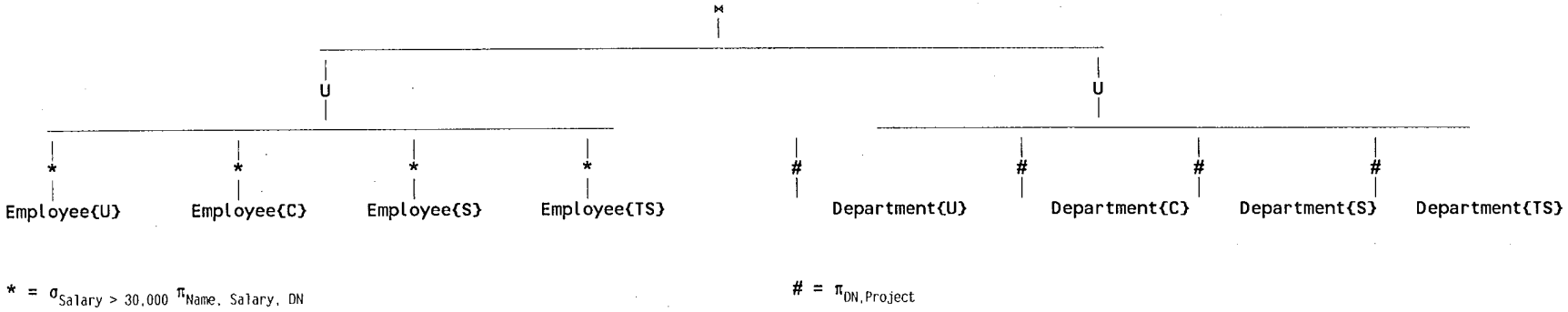
$R_{1,l} = \pi_{A1,Cl}$ with $l = C_1$
$R_{j,l} = \pi_{A1,Cl,Ai,Ci}$ with $j = A_i$ and $l = C_i \ \forall\ 2 \le i \le n$

where A_i is the *ith* attribute of R, and C_i is the classification of the *ith* attribute in R.

Query Tree Form



Distributed Sub-Queries: Natural Join after Union



Secure Distributed Sub-Queries: Natural Join after Union

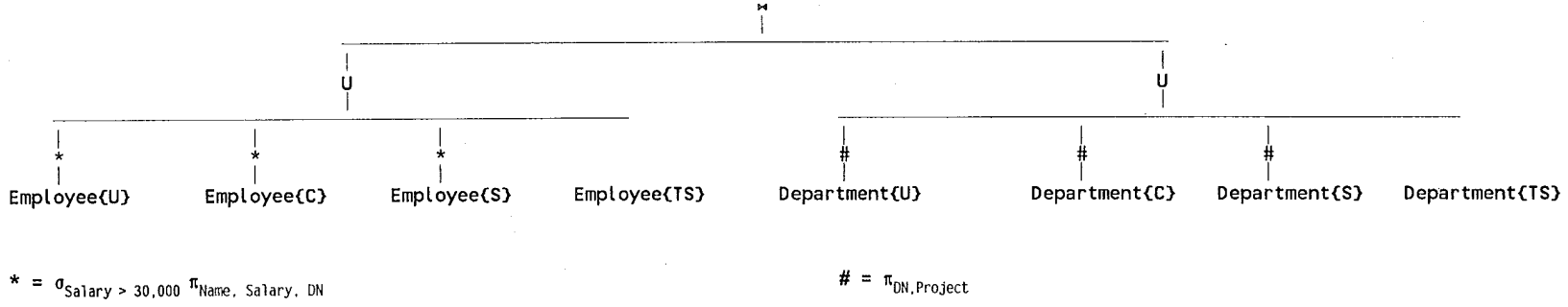


Figure 3
Secure Query Process: Tuple-Level Granularity

The formula -- $R_{1,l} = \pi_{A1,Cl}$ -- translate as follows: A subrelation primary key fragment will be generated by projecting the primary key and its classification attribute from each tuple at security classification *l*. The remaining formula -- $R_{j,l} = \pi_{A1,Cl,Ai,Ci}$ -- translate as follows: A subrelation attribute fragment will be generated by projecting a set containing a primary key attribute pair and one or more non-key attribute pair(s) for each non-key attribute in the tuple (the pairs consist of an attribute and its classification attribute).

The recovery algorithm, the inverse of F, uses the natural join operation to combine the single-level fragments of each relation into the original multi-level relation as follows:

$$R = \bowtie (R_{j,l}) \qquad \forall\ 1 \le j \le n \text{ and } l \in K$$

The symbol K stands for the set K which contains all of the security levels in the lattice, e.g., {U,C,S,TS}.)

When attribute-level labelling is used, each relational algebra expression is transformed into a safe expression with respect to a subject S logged-in at L(S) by replacing each multi-level relation in the expression with the natural join of those fragments of the relation that are dominated by L(S).

The following example demonstrates the above decomposition and recovery schemes. Assume that an instance of the Employee relation (R) is as follows:

Employee (R):

SSN	C_1	Name	C_2	Salary	C_3	Age	C_4	DN	C_5	TC
001	U	Able	C	20000	S	33	TS	C100	U	TS
002	U	Bond	C	45000	S	36	TS	C200	U	TS
003	U	Coe	C	70000	S	43	TS	C300	U	TS
004	U	Drake	C	135000	S	56	TS	C400	U	TS

The relation is decomposed as follows:

$D_{1,l} = \pi_{A1,Cl}$ with $l = C_1$
$D_{j,l} = \pi_{A1,Cl,Ai,Ci}$ with $j = A_i$ and $l = C_i$, $\forall\ 2 \le i \le n$

Note: the decomposed relation (single-level fragment) is represented by $D_{j,l}$ and the tuple classification attribute TC is dropped -- it is not required in a single-level fragment.

$D_{1,U}$

SSN	C_1
001	U
002	U
003	U
004	U

$D_{2,C}$

SSN	C_1	Name	C_2
001	U	Able	C
002	U	Bond	C
003	U	Coe	C
004	U	Drake	C

$D_{3,S}$

SSN	C_1	Salary	C_3
001	U	20000	S
002	U	45000	S
003	U	70000	S
004	U	135000	S

$D_{4,TS}$

SSN	C_1	Age	C_2
001	U	33	TS
002	U	36	TS
003	U	43	TS
004	U	56	TS

$D_{5,U}$

SSN	C_1	DN	C_4
001	U	C100	U
002	U	C200	U
003	U	C300	U
004	U	C400	U

The following formula recovers the relation R for a user logged-in at the Secret level:

$R_S = \bowtie (R_{j,l}), \quad \forall\ 1 \leq j \leq n$ and $L(S) \geq l$, where TC = max(l)

Max(l) in this example refers to the highest rank of any security classification found in $R_{j,l}$.

R_S:

SSN	C_1	Name	C_2	Salary	C_3	DN	C_4	TC
001	U	Able	C	20000	S	C100	U	S
002	U	Bond	C	45000	S	C200	U	S
003	U	Coe	C	70000	S	C300	U	S
004	U	Drake	C	135000	S	C400	U	S

In the examples in this section, the Employee relation defined above will serve as the first relation (R relation). The Department relation (or the W relation) will serve as the second relation. The Department relation has attributes DN, Dname, Project, and Manager (where DN is the primary key). The tuple classification attribute for the Department relations will be designated TE.

Department:

DN	C_1	Dname	C_2	Project	C_3	Manager	C_4	TE
C100	U	Computer	S	Software	TS	Smith	C	TS
C200	U	Crypto	S	Sneaky	TS	Jones	C	TS
C300	U	Foreign	S	Spying	TS	Bond	C	TS
C400	U	Nuclear	S	Weapon	TS	Green	C	TS

The Department relation is decomposed and recovered in the same manner as the Employee relation. The following sections demonstrate how and when secure query optimization is possible for the three relational algebra operations: Selection, Projection and Natural join.

The recovery technique above took the Natural-Join of all single-level fragments that are dominated by the level of the subject L(S) and then performed the query on the resultant relation. As was the case with tuple-level granularity, it is sometimes more efficient to decompose the query and send the resultant sub-queries to each of the fragments. The following sections demonstrate how and when secure query optimization is possible for the three relational algebra operations: Selection, Projection, and Natural Join.

3.3.1 Selection. The following formula states that to perform a secure selection operation (σ_F) on a multi-level relation R, is equivalent (->) to performing a selection operation over the natural join of all the decomposed single-level subrelations ($D_{j,l}$) which are dominated by the security level of the user [L(S)] which in turn is equivalent to taking the natural join after the selection operation has been performed on each subrelation $D_{j,l}$.

$$\sigma(\theta)\, R_{max(l)} \rightarrow \sigma(\theta)\, [\bowtie \{R_{j,l}\}],\ \forall\ 1 \le j \le n \text{ and } L(S) \ge l, \text{ where } TC = \max(l)$$

or equivalently by the distributivity of selections over joins,

$$\sigma(\theta)\, R \rightarrow \bowtie \{\sigma(\theta)\, R_{j,l}\}$$

Selection is distributive with join operations as long as certain necessary and sufficient conditions apply. In this case: SNC:$\exists\, F_1, F_2 : (F = F_1 \wedge F_2) \wedge (Attr(F_1) \subseteq Attr(R_{j,l})) \wedge (Attr(F_2) \subseteq Attr(R_{j+1,l}))$) [CerPel84].

The subscript j refers to the attribute being recovered (the A_j th or B_j th attribute). The subscript l refers to both

the security level of the single-level fragment the *ith* attribute is being recovered from and the level of that *ith* attribute. The resulting relation $R_{max(l)}$ is secure since all of the subrelations used to form it were dominated by L(S).

3.3.2 Projection. As the following formula states, a secure projection operation against a multilevel relation (R) is equivalent to making the projection after the natural join of all of the single-level subrelations dominated by L(S). (It is assumed the query optimizer will eliminate those fragments that are not required for the answer, i.e., those attributes not contained in the attribute list $\{A_h, \ldots A_m\}$)

$\pi (A_h,\ldots,A_m) R \rightarrow \pi (A_h,\ldots,A_m) [\bowtie \{R_{j,l}\}]$

$\forall\ 1 \leq h \leq n,\ 1 \leq m \leq n,\ 1 \leq j \leq n$, and $L(S) \geq l$, where $TC = \max(l)$

or equivalently by the distributivity of projections over joins

$\pi (A_h,\ldots,A_m) R \rightarrow \bowtie \{\pi(A_h,\ldots,A_m) R_{j,l}\}$

Projection is distributive over join provided all attributes that are part of the join criteria are contained in the attribute list, i.e., $Attr(join) \subseteq Attr(A_h, \ldots, A_m)$.) [CerPel84].

3.3.3 Natural Join

As the following formula indicates, the secure natural join of two multilevel relations, R and W, is performed after the natural join of all subrelations in R and W which are dominated by the security level L(S).

$R \bowtie W \rightarrow [\bowtie \{R_{j,l}\}] \bowtie [\bowtie \{W_{j,l}\}]$, where $1 \leq j \leq n$, and $L(S) \geq l$, where $TF = \max(l)$

Equivalently, distributing the natural join over the natural join

$R \bowtie W \rightarrow \bowtie [R_{j,l} \bowtie W_{j,l}]$

3.3.4. Secure Query Process Against Attribute-Level Granularity

Figure 4 shows the secure query process for queries made against attribute-level granularity for a query that combines the Selection, Projection, and Natural Join rules given above. Assume the user is logged-in at the secret level.

Figure 5 shows the secure query response to the query in Figure 4.

Figure 6 shows the secure query process in query tree form. This form illustrates the pruning process by showing the edges removed to all fragements whose security level is not dominated by the level of the subject.

SELECT Name, Salary, Age, DN, Dname
FROM Employee, Department
Where Salary > 30,000

$Q_S = \sigma_{Salary > 30,000}\, \pi_{Name, Salary, Age, DN, Dname}$ Employee $\bowtie$ Department

-> $\sigma_{Salary > 30,000}\, \pi_{Name, Salary, Age, DN}\{\bowtie Employee_{j,l}\} \bowtie \pi_{DN,Dname}\{\bowtie Department_{j,l}\}$

which equals: $\sigma_{Salary > 30,000}\, \pi_{Name, Salary, Age,DN}$ ($Employee_{2,C} \bowtie Employee_{3,S} \bowtie Employee_{4,TS} \bowtie Employee_{5,U}$) $\bowtie$ $\pi_{DN,Dname,}$ ($Department_{2,S}$)

which is equivalent to: ($\pi_{SSN,Name}$ $Employee_{2,C}$ $\bowtie$ $\sigma_{Salary > 30,000}\, \pi_{SSN, Salary}$ $Employee_{3,S}$ $\bowtie$ $\pi_{SSN,Age}$ $Employee_{4,TS}$ $\bowtie$ $\pi_{SSN, DN}$ $Employee_{5,U}$) $\bowtie$ ($\pi_{DN,Dname}$ $Department_{2,S}$)

Figure 4
Secure Query Process Against Attribute-Level Granularity

$\sigma_{Salary > 30,000}\, \pi_{Name, Salary, DN, Dname}$ Employee $\bowtie$ Department

Name	C_1	Salary	C_2	R.DN	C_3	Dname	C_4	TF
Bond	C	45000	S	C200	U	Crypto	S	S
Coe	C	70000	S	C300	U	Foreign	S	S
Drake	C	135000	S	C400	U	Nuclear	S	S

Figure 5
Secure Query Response for Attribute-Level Granularity

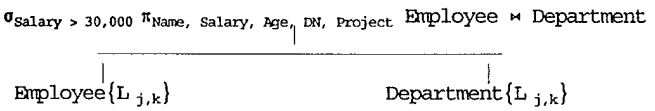
3.4 Element-Level Granularity

When element-level labelling is used, each attribute value A_j in each relation has an associated classification C_i. To guard against unauthorized access to data, each relation is transformed into single-level fragments using a decomposition algorithm F. The decomposition algorithm used by Jajodia and Sandhu is used for the examples in this section [JajSan91]. Each relation has the form $R(A_1,C_1,...,A_n,C_n,TC)$ where C_i is the level of the A_i attribute and TC is the tuple classification. In this scheme, assume that the primary key is designated as A_1 and its classification level is C_1. A multilevel relation is stored as a collection of single-level base relations of the form:

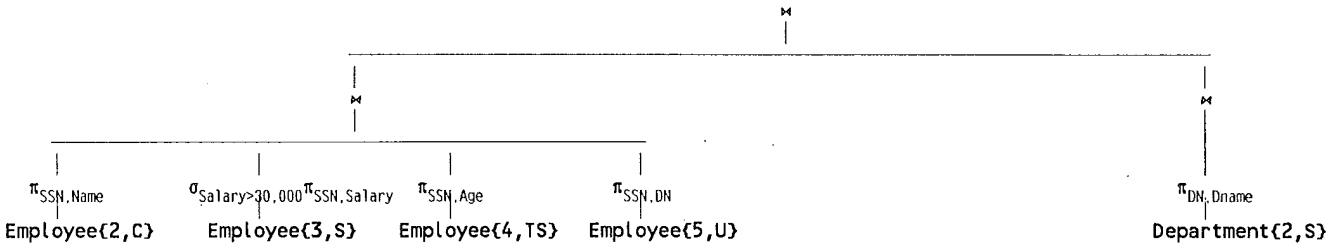
$D_C(A_1,C_1,...,A_n,C_n)$

where A_1 through A_n are the attributes of the relation, C_1 through C_n are the classifications of their respective attributes, and *c* is the access class of the base relation.

Query Tree Form



Distributed Sub-Queries: Natural Join



Secure Distributed Sub-Queries: Natural Join

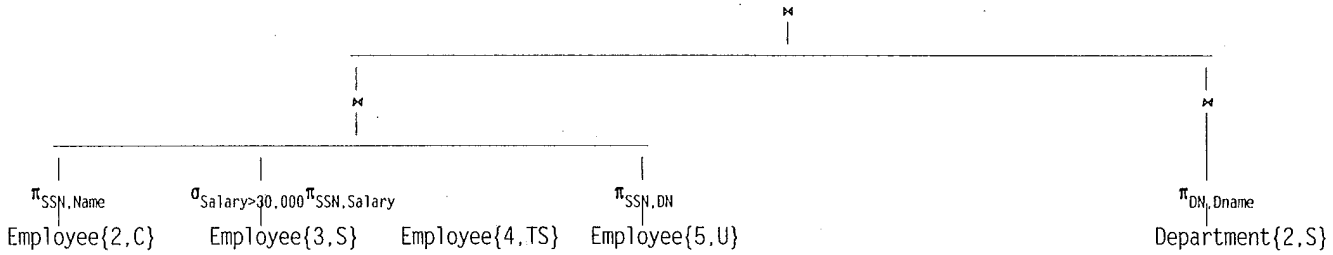


Figure 6
Secure Query Process: Attribute-Level Granularity

There is one base relation for each access class *c* in the lattice. A user logged-in at any given level *c* always sees and interacts with a relation Rc which is recovered from the base relations that the *c*-user (a user at clearance level *c*) is entitled to see. Readers interested in more detail are invited to see [JajSan91].

Relations are decomposed into their corresponding single-level base relations as follows:

$$R(A_1,C_1,...,A_n,C_n,TC) \rightarrow D_j(A_1,C_j,A_2,C_j,...,A_n,C_j)$$

where $l \leq j \leq h$ and A_i = null if $C_i <> C_j$ and takes the value from R otherwise.

The following example shows how the above decomposition is used. Given the Employee Relation (R):

R

SSN	C_1	Name	C_2	Salary	C_3	DN	C_4	TC
001	U	Able	S	20000	U	C100	U	S
002	U	Bond	U	45000	S	C200	U	S
003	C	Coe	C	70000	S	C300	C	S
004	C	Drake	C	135000	TS	C400	C	TS

The following example shows how the relation R would be decomposed using the above decomposition scheme:

D_U

SSN	C_1	Name	C_2	Salary	C_3	DN	C_4
001	U	null	U	20000	U	C100	U
002	U	Bond	U	null	U	C200	U

D_C

SSN	C_1	Name	C_2	Salary	C_3	DN	C_4
003	C	Coe	C	null	C	C300	C
004	C	Drake	C	null	C	C400	C

D_S

SSN	C_1	Name	C_2	Salary	C_3	DN	C_4
001	U	Able	S	?	U	?	U
002	U	?	U	45000	S	?	U
003	C	?	C	70000	S	?	C

D_{TS}

SSN	C_1	Name	C_2	Salary	C_3	DN	C_4
004	C	?	C	135000	TS	?	C

To recover the relation R*c* for a *c*-user [JajSan91]:

1. Take the union of all lower base relations:

   $\cup_{c' \leq c} D_{c'}$

   Add to each tuple in the result its tuple class TC (t[TC] = lub $\{t[C_i]: i=1...n\}$). The result is R_c.

2. Remove deleted Keys from R_c:

   Let t_1 contained in R_c be such that $t_1[C_1] < c$ and R_c does not contain a t_2 such that $t_2[A_1,C_1] = t_1[A_1,C_1]$ and $t_2[TC] = t_1[C_1]$. Then we delete t_1 from R_c. If $t_1[TC] = c$, then we delete t_1 from D_c as well.

3. Apply the ?-replacement rule to R_c:

   Let t be a tuple in R_c with $t[A_k]$ = '?'.

   a. If there is a tuple u contained in R_c with $u[A_1,C_1] = t[A_1,C_1]$ and $TC[u] = t[C_k]$, replace '?' in $t[A_k]$ by $u[A_k]$.

   b. If there does not exist a tuple u contained in R_c with $u[A_1,C_1] = t[A_1,C_1]$ and $TC[u] = t[C_k]$, replace '?' by 'null' in $t[A_k]$.

4. Make R_c subsumption-free:

   Remove all tuples s such that for some t contained in R_c and for all $i=1..n$ either $t[A_i,s_i] = s[A_i,s_i]$ or $t[A_i] <> \text{null}$ and $s[A_i] = \text{null}$.

The following example shows the use of this recovery algorithm for a user logged-in at the secret level:

R_S

SSN	C_1	Name	C_2	Salary	C_3	DN	C_4	TC
001	U	Able	S	20000	U	C100	U	S
002	U	Bond	U	45000	S	C200	U	S
003	C	Coe	C	70000	S	C300	C	S
004	C	Drake	C	null	C	C400	C	C

The above recovery technique uses the union operation to combine all single-level fragments that are dominated by the level of the subject L(S) and then performs the query on the resulting relation. As was the case with both tuple-level and attribute-level granularity, it is sometimes more efficient to decompose the query and send the resulting sub-queries to each of the fragments.

For the examples in the remainder of this section, the Employee relation defined above will serve as the first relation (R relation). The Department relation (or the W relation) will serve as the second relation. The

Department relation has the same attributes as described in the previous sections.

W

DN	C_1	Dname	C_2	Project	C_3	Manager	C_4	TE
C100	U	Computer	C	Software	C	Smith	U	C
C200	U	Crypto	S	Sneaky	U	Jones	U	S
C300	C	Foreign	C	Spying	S	Bond	S	S
C400	C	Nuclear	TS	Weapon	TS	Green	C	TS

The Department relation is decomposed and recovered the same way as the Employee relation. The next sections demonstrate the query modification process for element-level granularity performed for the three relational algebra operations: Selection, Projection, and Natural Join. The transformation will be shown using the Jajodai-Sandhu algorithm which will be represented by the following formula:

$F^{-1} R_{i:L(S)}$ where F^{-1} indicates the four-step recovery algorithms is being applied to recover the relation R at level i=L(S).

3.4.1 Selection. The following formula states that to perform a secure selection operation (σ_F) on a multi-level relation R, is equivalent (->) to performing a selection operation after the union of all the decomposed single-level subrelations which are dominated L(S).

$\sigma(\Theta)\ R \rightarrow \sigma(\Theta)\ F^{-1} R_{i:L(S)}$ where $l \leq i \leq L(S)$.

3.4.2 Projection. The following formula states that a secure projection operation against a multilevel relation R is equivalent to making the projection against the multi-level relation $R_{L(S)}$ recovered from all of the single level subrelations dominated by L(S):

$\pi\ (A_j,...,A_m)\ R \rightarrow \pi\ (A_j,...,A_m)\ [F^{-1} R_{i:L(S)}]$ where $j \geq 1$ and $m \leq n$.

or equivalently by the distributivity of projection over union

$\pi\ (A_j,...,A_m)\ R \rightarrow [F^{-1} R_{i:L(S)}]\ \pi\ (A_j,...,A_m)\ R_{i:L(S)}$ where $j \geq 1$ and $m \leq n$.

3.4.3 Natural Join. The formula for arriving at the natural join is:

$R \bowtie W \rightarrow [\ F^{-1} R_{i:L(S)}\ ] \bowtie [\ F^{-1} W_{i:L(S)}\ ]$

3.4.4. Secure Query Process Against Element-Level Granularity

Figure 7 shows the secure query process for queries made against element-level granularity for a user logged-in at the secret level.

Figure 8 shows the secure query response to the query in Figure 7.

SELECT Name, Salary, DN, Dname
FROM Employee, Department
Where Salary > 30,000

$Q_S = \sigma_{Salary > 30,000}\, \pi_{Name, Salary, DN, Dname}\, \{F^{-1}\ Employee_{i:L(S)}\} \bowtie \{F^{-1}\ Department_{i:L(S)}\}$

which is equivalent to: $\sigma_{Salary > 30,000}\, \pi_{Name, Salary, DN}\, \{F^{-1}\ Employee\} \bowtie \pi_{DN,Dname}\, \{F^{-1}\ Department\}$

Figure 7
Secure Query Process Against Element-Level Granularity

$\sigma_{Salary > 30,000}\, \pi_{Name, Salary, DN, Dname}\, \{F^{-1}\ Employee_{i:L(S)}\} \bowtie \{F^{-1}\ Department_{i:L(S)}\}$

Name	C_1	Salary	C_2	R.DN	C_3	Dname	C_4	TF
Bond	U	45000	S	C200	U	Crypto	S	S
Coe	C	70000	S	C300	C	Foreign	C	S

Figure 8
Secure Query Response for Attribute-Level Granularity

Figure 9 illustrates the secure query process for element-level granularity in query tree form.

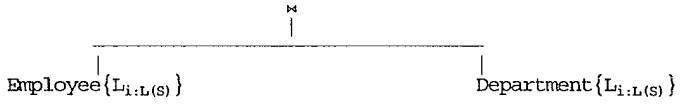
4. CONCLUSION

Secure query modification for multi-level databases which are implemented as kernelized architectures can be implemented using a divide and conquer approach.. Each of the single-level fragments is a subrelation of the multi-level relation. This is directly analogous to a distributed database. The distribution algorithm used is based on the security levels involved and the granularity of the labeling scheme.

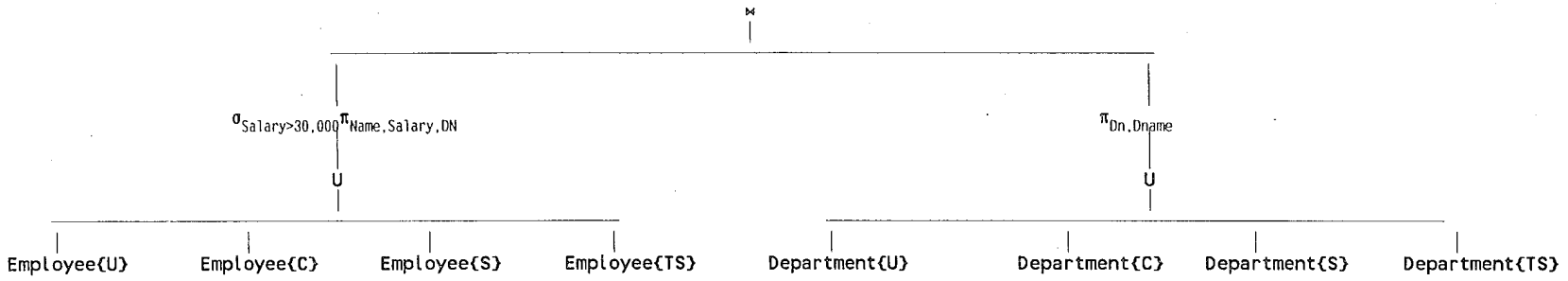
In conclusion, by using well know query optimization techniques developed for distributed database is should be possible to modify the query processor to parse the query fragments generated during the optimization step to eliminate any queries to fragments whose security level is not dominated by the level of the user. The resultant answer will be secure since no subrelation not dominated by the level of the user are included in the modified output of the query optimizer. The query processing techniques developed for transforming global queries into query fragments for distributed databases can be used to efficiently and effectively implement security constraints in multi-level databases implemented as kernelized architectures. This has been demonstrated for the following levels of granularity: tuple, attribute, and element.

When multi-level relations are decomposed into single-level fragments based on tuple-level granularity, each of the resultant subrelations is a horizontal fragment of the parent relations. Query transformation techniques from distributed data processing can be used to convert global queries against the multi-level parent relation into a set of sub-queries against each of the horizontal fragments. These horizontal fragments are disjoint and can be reconstructed into the parent relation through the use of the union operation. In the case of horizontal fragment schemes (tuple-level security), this process makes a copy of the original query, sends each of

Query Tree Form



Distributed Sub-Queries (Natural Join after Union)



Secure Distributed Sub-Queries (Natural Join after Union)

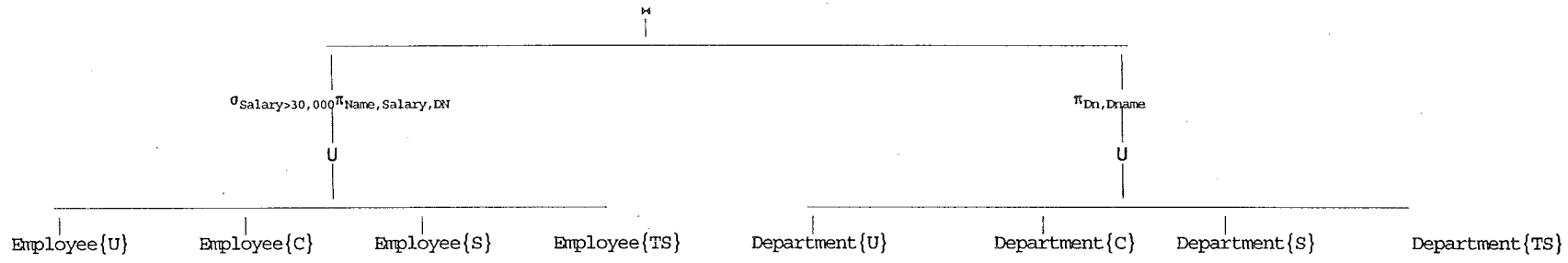


Figure 9
Secure Query Process: Element-Level Granularity

the copies to the appropriate subrelation (fragment), and combines the resultant query responses into a global response. This paper has demonstrated that the query fragment list can be parsed. Any query fragments which would violate security constraints (by accessing a subrelation not dominated by the security level of the subquery fragment) can be deleted from the query fragment list. The resultant query fragment list is secure since none of the remaining fragments reference any subrelations that are not dominated by them.

When multi-level relations are decomposed into single-level fragments based on attribute-level granularity, each of the resultant subrelations is a vertical fragment of the parent relations. Each of the vertical fragments consists of a couple [primary key, {attribute list}]. A copy of the apparent primary key is placed in each vertical fragment so that the join operation can be used to reconstruct the parent relation. Each attribute, except for the apparent primary key is placed in exactly one vertical fragment. The attribute list contains the names of all those attributes classified at each distinct level in the lattice. Conversion of the global query is more complicated than in the tuple-level case because different operations are sent to different query fragments. (It would be incorrect and inefficient to send a selection based on the value of the name attribute to any fragments except those containing the name attribute values.) After the conversion process is complete, a list of query fragments is produced. This list can be parsed as mentioned previously. All query fragments directed at subrelations not dominated by them would be deleted. The resultant query list would be secure since no query fragments would remain that would be against subrelations not dominated by the query fragments.

When multi-level relations are decomposed into single-level fragments based on element-level labeling, the resultant subrelations are equivalent to a mixed fragmentation scheme. Each mixed fragment consists of a couple [primary key, {attribute list}]. As in the treatment of vertical fragments, the apparent primary key is included in each subrelation so that the parent relation may be reconstructed. Depending on the reconstruction algorithm used, either the join or union operation is used to perform the reconstruction. The attribute list contains the names of all those attributes classified at each level in the lattice. In the case of element-level labeling, a given attribute may be classified at any level in the lattice. This increases the number of subrelations and makes both the query process and the reconstruction process more complicated. However, the steps are still the same. The global query is transformed into a number of query fragments which are placed on the query list. The query list is then parsed, and any query fragment is rejected if it does not dominate the subrelation to which it is directed. The resultant query fragment list is safe since no query fragments remain which do not dominate the subrelations against which they are directed.

Implementing security constraints in MLS DBMS continues to be a dynamic area of research. Many different models have been proposed. None of them are beyond the prototype stage. While demonstrating significant potential as a means to produce safe queries and meaningful responses, this paper also points to a number of areas of additional research which must be done before the full utility of the proposed model is understood. Additional research needs to be conducted on the overall cost and benefit of the technique proposed in this paper, as well as on ways to most efficiently carry out the approach.

In particular, research is needed on efficient techniques to parse the fragment list during the deletion process and the potential impact of deleting query fragments. It is not clear from this research if the savings from not processing the query fragments which are deleted will offset the cost of implementing this strategy. How much additional cost will be introduced by the query deletion phase? How much saving will be generated by not making the queries that would violate security constraints? A prototype system should be developed so that these implementation issues can be explored and processing impact determined.

BIBLIOGRAPHY

[CerAL83] Ceri, S; Navathe, B; Wiederhold, G (1983): Distribution Design of Logical Database Schemas. IEEE-TSE SE:-9:4, 487-503.

[CerAL82] Ceri, S; Negri, M; Pelagatti, G (1982): Horizontal Data Partitioning in Database Design. ACM SIGMOD, 128-136.

[CerPel83] Ceri, S; Pelagatti, G (1983): Correctness of Query Execution Strategies in Distributed Databases. ACM Transactions on Database Systems 8:4, 577-607.

[CerPel84] Ceri, S; Pelagatti, G (1984): Distributed Database Principles and Systems. McGraw-Hill, New York.

[ChaChe80] Chang, SK; Cheng, WH (1980): A Methodology for Structured Database Decomposition. IEEE-TSE SE-6:2, 205-218.

[CMDM83] Committee on Multilevel Data Management (1983): Multilevel Data Management Security. Technical Report, Air Force Studies Board, National Research Council, 1983.

[Date90] Date, CJ (1990): An Introduction To Database Systems. Fifth ed. Vol. I. Addison-Wesely, Reading, Massachusetts. 854 pages.

[Denn85] Denning, DE (1985): Commutative Filters for Reducing Inference Threats in Multilevel Database Systems. Proceedings IEEE Symposium on Research in Security and Privacy, 1985, 134-146.

[DenAL86] Denning, DE; Lunt, TF; Neumann, PG; Schell, RR; Heckman, M; Shockley, W (Nov. 1986): Security policy and interpretation for a class A1 multilevel secure relational database system., .

[DenAL87a] Denning, DE; Akl, SG; Heckman, M; Lunt, TF; Morgenstern, M; Neumann, PG; Schell, RR (Feb. 1987): Views for multilevel database security. IEEE Trans. on Software Engineering. SE-13(2), 129-140.

[DenAL87b] Denning, DE; Lunt, TF; Schell, RR; Heckman, M; Schockley, W (1987): A Multilevel Relational Data Model. Proc. IEEE Symposium on Security and Privacy, 220-234.

[DenAL87c] Denning, DE; Lunt, TF; Schell, RR; Heckman, M; Shockley, WR (1987): A Multilevel Relational Data Model. Proc. IEEE Symposium on Security and Privacy, 220-234.

[DenAL88] Denning, DE; Lunt, TF; Schell, RR; Shockley, WR; Heckman, M (1988): The SeaView Security Model. Proc. IEEE Symposium on Security and Privacy, 218-233.

[Garv86] Garvey, C (1986): Multilevel Data Store Design (MLDS). Technical Report, TRW Defense Systems Group, 1986.

[Groh76] Grohn, MJ (June 1976): A Model of a Protected Data Management System. Technical Report ESD-TR-76-289, I. P. Sharp Accoc. Ltd., June 1976.

[HinSch75] Hinke, TH; Schaefer, M (1975): Secure Data Management System. Technical Report RADC-TR-75-266, System Development Corp., Nov. 1975.

[JajSan90a] Jajodia, S; Sandhu, R (1990): Polyinstantiation Integrity in Multilevel Relations. Proc. IEEE Symposium on Security and Privacy, 104-115.

[JajSan90b] Jajodia, S; Sandhu, R (1990): Polyinstatiation Integrity in Multilevel Relations Revisited., Database Security IV: Status and Prospects, Jajodia, S. and Landwehr, C. (editors), North-Holland

[JajSan91] Jajodia, S; Sandhu, R (1991): A Novel Decomposition of Multilevel Relations Into Single-Level Relations. Proceedings IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 20-22, 1991, 300-313.

[KeeAL89] Keefe, TF; Thuraisingham, MB; Tsai, WT (1989): Secure Query-Processing Strategies. IEEE 1989, 63-70.

[LunAL88] Lunt, TF; Schell, RR; Shockley, WR; Heckman, M; Warren, D (1988): A Near-Term Design for the SeaView Multilevel Database System. Proceedings IEEE Symposium on Security and Privacy, April 1988.

[Mart94] Martin, ML: Improving Security In Multi-Level database Management Systems Through The Use of Query Modification. Ph.D. Dissertation, George Mason University, 247.

[NCSC90] National Computer Security Center (1990): Trusted Database Management System Interpretation of the Trusted Computer Systems Evaluation Criteria. Report NCSG-TG-021 Version-1.

Vita

Michael L. Martin is a Computer Scientist at the Center for Software at the Defense Information Systems Agency [DISA].* He has had more than 10 years of experience working on Defense Department database interoperability and data administration issues. Prior to joining the Defense Department, Dr. Martin worked on database management issues at the Department of Health and Human Services in the Social Security Administration and the Health Care Financing Administration.

He received his Ph.D. in Information Technology from George Mason University in 1995. His dissertation was entitled *Improving Security in Multi-Level Database Management Systems through the Use of Query Modification.* Dr. Martin served as a Command, Control, Communications and Intelligence Research Fellow at George Mason University under a fellowship supported by DISA. He received a Masters of Science in Computer Science from The Johns Hopkins University, Baltimore, Maryland in 1985 and a Masters of Business Administration from St. Louis University, St. Louis, Missouri in 1972. He received his Bachelor of Arts in Psychology from the University of Montana at Missoula, Montana, in 1968.

*This paper represents the views of the author, not DISA or any other agency in the federal government.

High Assurance MLS Database Applications

by Tom Haigh, Dick O'Brien and Dan Thomsen
Secure Computing Corporation
2675 Long Lake Road
Roseville, MN 55113

Under the LOCK DBMS program, funded by the Air Force's Rome Laboratory, Secure Computing Corporation (SCC) has implemented Trusted ORACLE V7 on the Secure Network Server (SNS) system. The SNS is a highly assured multi-level secure (MLS) network server product based on the LOCK prototype. It is the network server component of the MISSI program and provides the means for securely connecting networks at different classification levels. Thus, LOCK DBMS is the first high assurance DBMS that can support highly secure, MLS database applications that share information among networks at different classification levels. This paper describes the LOCK DBMS system and some of the MLS applications that are being developed using it.

The applications that are discussed in the paper include:

- Connection of legacy databases at different levels
  Users are able to update information at their security level and have this information immediately visible to higher level users and their applications. There are two approaches: either storing shared data in the LOCK DBMS or allowing assured queries from higher levels to lower levels. In the first approach, information can be directly entered by low-level users into the database on the LOCK DBMS, or information in databases on the low-level network can be pulled into the LOCK DBMS system using database applications, running on the SNS at the lower level, that periodically retrieve information from the network databases. The second approach uses assured stored procedures running on the SNS that are initiated by a high-level user and that access the lower level databases.
- MLS directory server database
  An MLS X.500 directory service, implemented using LOCK DBMS on the SNS system and including support for a multilevel directory database, provides the ability to define directory entries that have attributes stored at different levels. An entry with classified attributes can be partitioned into lower-level versions, that have only the unclassified attributes, and higher-level versions, that contain the complete information on the entry. Users see only the information about the entry that they are cleared to see.
- Dataguard database
  The SNS has integrated FORTEZZA support to provide cryptographic services. These services allow the SNS to be used as a dataguard system; that

is, the SNS serves as a front-end to a system high network on which documents at different security classifications are stored. By cryptographically sealing the documents before they are entered into the system high network and then checking the seals before they are released, the dataguard ensures that documents do not become "contaminated" with classified information while stored on the system high network. The LOCK DBMS system is used to provide a document index and to store the document identifiers and associated cryptographic seals.

A significant feature of the SNS is its support for type enforcement, a mechanism that provides the ability to protect the integrity and privacy of data in a manner not found on other network servers. Each of these applications also makes use of the type enforcement mechanism to implement highly assured, role-based access control policies for protecting the databases on the SNS.

The paper is organized as follows. In Section 1, a description of the LOCK DBMS system is given. Then in Sections 2, 3, and 4, each of the MLS DBMS applications is described in more detail.

1. The High Assurance LOCK DBMS System

The LOCK DBMS system is a high assurance, state-of-the-art Trusted DBMS (TDBMS) prototype running as a stand-alone application on the SNS platform. The SNS product is based on the LOCK prototype, which was designed to meet the Class A1 requirements of the Trusted Computer Systems Evaluation Criteria (TCSEC). The commercial TDBMS that is being used for LOCK DBMS is Trusted Oracle Version 7. The LOCK DBMS system was successfully demonstrated at Rome Laboratory in June, 1994 and has been demonstrated at both the NSA and Rome Lab Tech Exchanges since then. Work is currently planned to enhance the usability of the system based on comments received from the Rome Lab evaluation. Initial design work on the network components has also been completed.

The LOCK DBMS system has the following features.

- It is based on the trusted computing base (TCB) subset paradigm described in the Trusted Database Interpretation of the TCSEC (TDI).

  In the LOCK DBMS design, there are two TCB subsets. The underlying subset is the high assurance SNS system. It handles all the Mandatory Access Control (MAC) enforcement, type enforcement, identification and authentication of the database users, and provides an underlying audit mechanism. Layered on top of the SNS TCB is the Trusted Oracle TDBMS. It provides DBMS named objects that correspond to tuples in a table and supports discretionary access control (DAC) and audit at the tuple level of granularity. There is a separate instance of the TDBMS executing at each level.

- It integrates the SNS type enforcement mechanism into the design to provide additional security and integrity features that advance the state-of-the-art in TDBMS technology.

  By defining special database domains and types, the database files are completely isolated from the rest of the system, and access to the files is restricted, in a mandatory manner, to the TDBMS subjects. The SNS type enforcement mechanism also allows roles to be implemented in a mandatory manner with high assurance.

- It uses a client/server architecture that offers high assurance and performance and allows for future development of a network database server.

  The system architecture is based on Trusted Oracle's client/server model that provides the best performance for the TCB subset design. There is only one TDBMS instance per level, rather than one TDBMS instance for each user application, so the number of TDBMS subjects on the system is greatly reduced, resulting in better performance. Since the LOCK DBMS system already uses the client/server model, it will be possible, with minimum effort, to extend LOCK DBMS to act as a database server for clients operating on networks that are connected to the SNS system.

- It provides state-of-the-art commercial DBMS functionality.

  Trusted ORACLE V7 provides full ANSI SQL compatibility, secure concurrency control for transactions at different levels, support for roles and triggers, extensions for dealing with labeled data and a variety of other modern DBMS features.

Current plans are to enhance the current LOCK DBMS prototype into a networked TDBMS server by adding the networking support available through Oracle's SQL*Net and Secure Network Services products.

2. Connecting Legacy Databases

Because the LOCK DBMS system is based on the high assurance SNS TCB, it provides the ability to connect legacy databases on different networks, at possibly different classification levels, in a secure manner. Two approaches are presented here:

- Distributed MLS Systems
- Secure Legacy Access.

Distributed MLS Systems

The LOCK DBMS system can be used to connect legacy databases on separate classified networks into a virtual classified distributed database. As Figure 1 indicates, on each network the legacy databases can be connected as a distributed database with

the LOCK DBMS system associated with that network. Between networks, all traffic is encrypted using Type 1 encryption. This allows the LOCK DBMS systems at each classification level to operate as a distributed database. The net effect is that legacy databases at the same classification level but on different networks can interoperate.

Since the LOCK DBMS system is based on Oracle V7, full distributed connectivity is only possible with other Oracle databases. However, tools exist that support sharing of data between heterogeneous systems, and these tools could be used on each network to connect non-Oracle DBMSs into the system.

Secure Legacy Access

Trusted filter processes that run on the SNS can be developed and assured to provide high assurance multilevel transactions. This allows special transactions to be defined that have the ability to execute at several levels and that can be verified to not leak classified information. Such transactions provide a secure approach to sharing and updating information between single-level databases on networks at different levels. In particular, as Figure 2 illustrates, users on higher level systems can access, and possibly update, information in legacy databases at lower levels. The particular transactions and type of access allowed could be carefully controlled using the SNS's type enforcement mechanism.

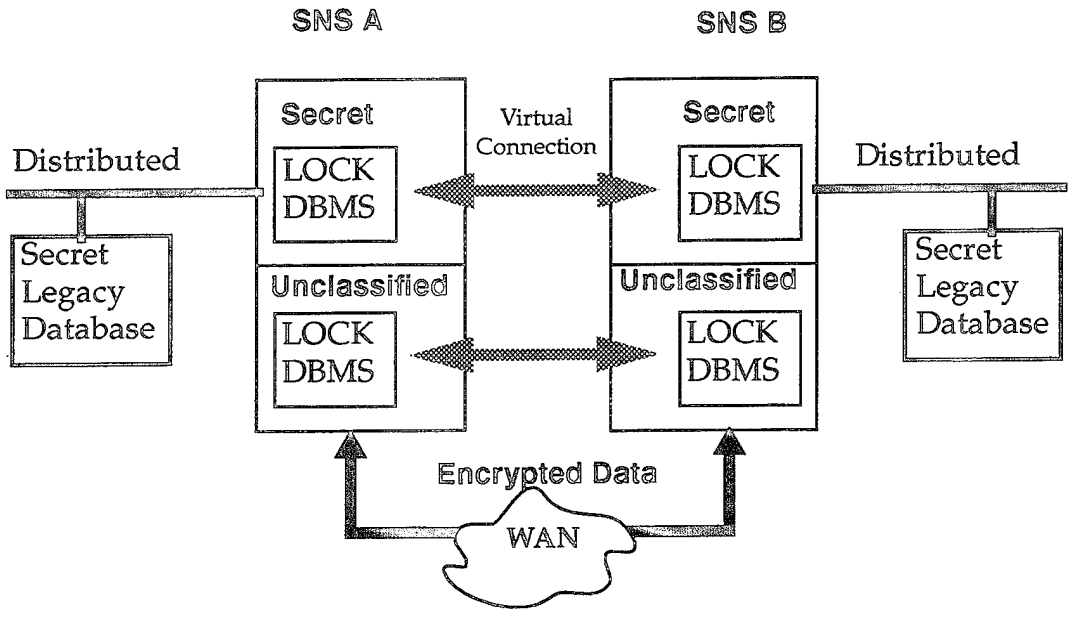


Figure 1: MLS Distributed Database System

LOCK DBMS on the SNS system is used to connect legacy systems on different networks in a distributed manner.

3. Secure X.500

X.500 is an international standard that defines an application-layer protocol for network directory services. The original standard was adopted in 1988 and extensions to the standard were added in 1993. The main components of an X.500 system (shown in Figure 3)) are:

- The directory information base (DIB)
- Directory system agents (DSAs)
- Directory user agents (DUAs).

The DIB is the database of information stored in the directory. It is logically organized in a hierarchical tree with each node in the tree representing a directory entry. Attached to each entry is a list of attributes that describe that entry. The DIB is managed by the DSAs, which act as directory servers to the DUAs, the directory clients. X.500 defines two protocols for accessing the information in the DIB: a directory access protocol (DAP) that DUAs use to talk to DSAs and a directory system protocol (DSP) that DSAs use to talk to other DSAs.

The basic purpose of an X.500 directory is to go beyond simple network name services, that can be used to map names to network addresses, to provide more extensive directory information on network entities such as users, hosts and other network resources. Directory information can be distributed across heterogeneous networks on a global scale and accessed from DUAs via the X.500 protocols. Some of the proposed uses of X.500 include providing directory services for national (and international) electronic mail, for telephone white pages and yellow pages, and for

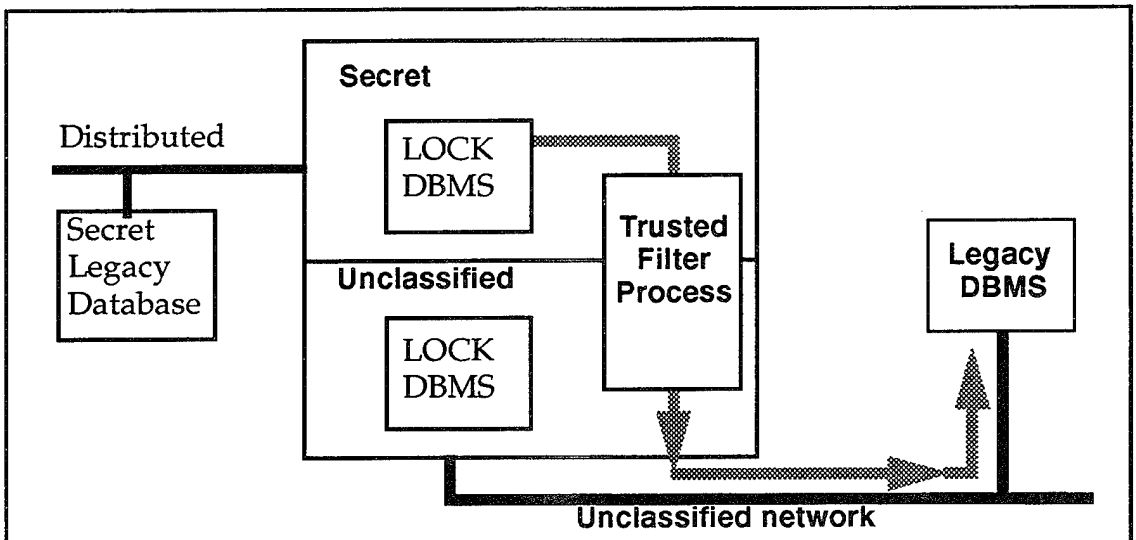


Figure 2: Accessing Legacy Databases Securely

Trusted filter processes on the SNS control which multilevel transactions are allowed.

corporate and government personnel directories. A high assurance, secure X.500 directory server is a major enabling technology for current and future efforts to provide secure, transparent interoperability between disparate networks as demonstrated in Figure 3. Such efforts include the National Information Infrastructure (NII) and the Defense Message System (DMS). (X.500 is an identified component of the DMS).

Other more specific uses have been proposed including using the DIB to store key certificates for public key cryptographic systems. This would be an especially appropriate service to provide on a high assurance system such as SCC's SNS. If the integrity of public key material were lost, it would be possible for a criminal to perpetrate fraud by substituting his own public key information in directory entries and then forging signatures of the organizations and individuals associated with those signatures. Thus, the directory must be maintained by authorized individuals utilizing programs that have been certified to make only the modifications indicated by the individual running the program.

The planned high-assurance, secure X.500 system on the SNS, using LOCK DBMS to implement the DIB, differs from a standard X.500 system in the following ways.

- It provides high assurance based on the underlying SNS platform. The directory can be protected by both traditional multilevel access controls and the SNS's special type enforcement for providing subsystem separation and

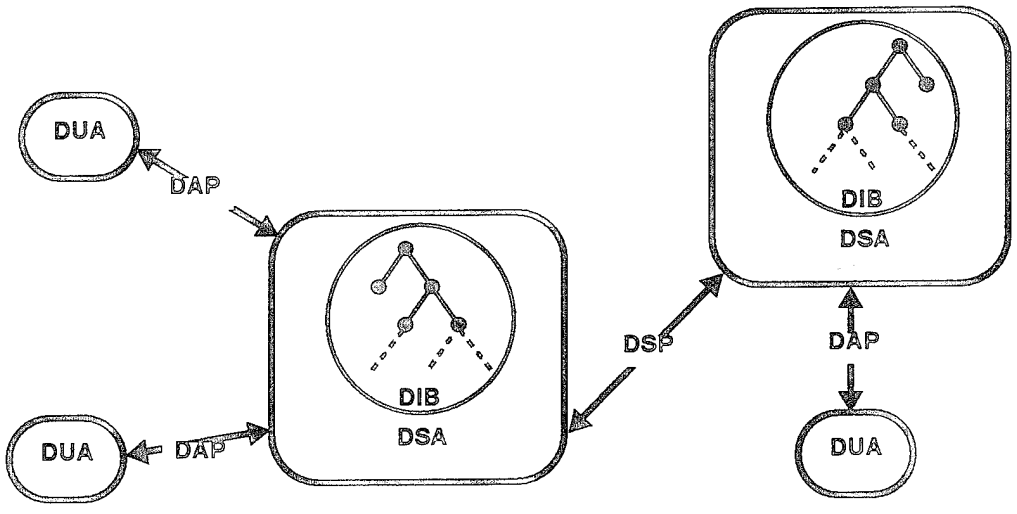


Figure 3: Components of an X.500 System

Users access the directory database from DUA clients running on their systems using the DAP protocol. The DSA acts as a directory server, maintaining the directory information in the DIB, responding to requests from the DUAs and, when necessary, forwarding requests to other DSAs, using the DSP protocol.

high integrity.

- It provides strong Identification and Authentication as defined in the X.509 standard using the FORTEZZA functionality.
- It allows classified attributes of an entity to be entered into the database, stored at the appropriate level, and protected from unauthorized access. That is, the DIB can be a multilevel database.
- It provides the ability for an X.500 directory to be multilevel in a secure manner. To be truly useful, a secure X.500 system must allow DUAs on a classified network to obtain information from DSAs on other networks. DUAs on a higher level can obtain information from DSAs at a possibly lower level by making a request to a DSA, operating at the same level of the DUA, on an MLS system and having that DSA request the information from the lower level DSA (via a process called chaining), using filters on the MLS system to downgrade the request. Rather than trust the DSA to do this

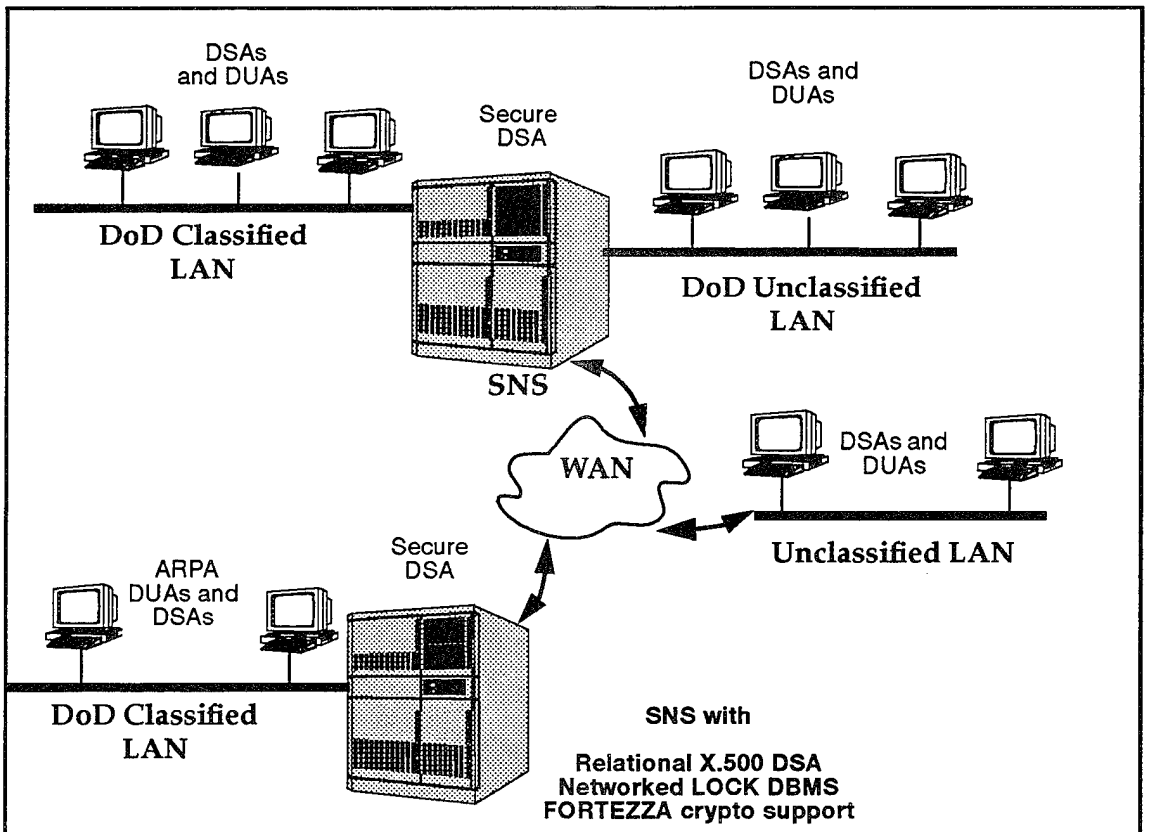


Figure 4: High Assurance, Secure X.500 System

The X.500 service is implemented using the high assurance LOCK DBMS system. Directory User Agent (DUA) clients at different levels can access directory information based on their levels. Secure Directory System Agents (DSAs) use LOCK DBMS to store multilevel data and can forward requests across a wide area network to other DSAs.

correctly, a reclassifier will pass the request down, verify and audit the request, and then verify and pass the response back up in level.

4. Dataguard Database

Document databases are currently being developed that involve acquiring, managing and distributing information in a classified electronic environment. Documents of differing classification are stored on a system high network with users on the network able to view the documents using standard graphical user interfaces. A major problem that such systems must address is how to release documents stored on the system high network in a secure yet cost-effective manner. If data is to be stored on and distributed from the document database, it must be carefully reviewed to ensure that information to be exported contains only the information authorized for export. Due to the large amount of data that will be in the document database and the possible technical content of the data, conducting adequate and timely manual reviews of documents to be distributed (to ensure no sensitive information is included) would be extremely difficult if not impossible.

SCC is developing the SNS Dataguard system to provide a solution to this problem that guarantees that documents cannot be modified, while stored in the document database, without detection. By providing a means for easily checking whether a document has been modified, the need for costly, difficult reviews is eliminated. A quick computerized check is done to determine if the document to be released has been modified in any way since it was entered into the database.

SCC's approach for detecting unauthorized modifications of documents in the document database is shown in Figure 5. There are two key elements.

- The first is the use of MISSI MOSAIC cryptography to allow modifications to documents in the document database to be easily detected. Each document is cryptographically sealed when it is entered into the system. Before releasing the document, the crypto seal is recalculated. If the document has been modified in anyway, the recalculated crypto seal will not match the original crypto seal.
- The second key element is the high assurance SNS platform where the documents can be sealed and the cryptographic seals stored. On an untrusted system no guarantee can be made that the document is not modified before the crypto seal is calculated. Malicious software may modify the document, by copying sensitive data into it, and then use the cryptographic key to recalculate the cryptographic seal. The SNS system thwarts these attacks using its type enforcement mechanism. Type enforcement allows "assured pipelines" to be created which ensure that all operations and checks are properly performed before a document is inserted

into or released from the system. Type enforcement also separates the cryptographic subsystem from untrusted user software.

Additional Features

A variety of additional features can be added to provide either additional capabili-

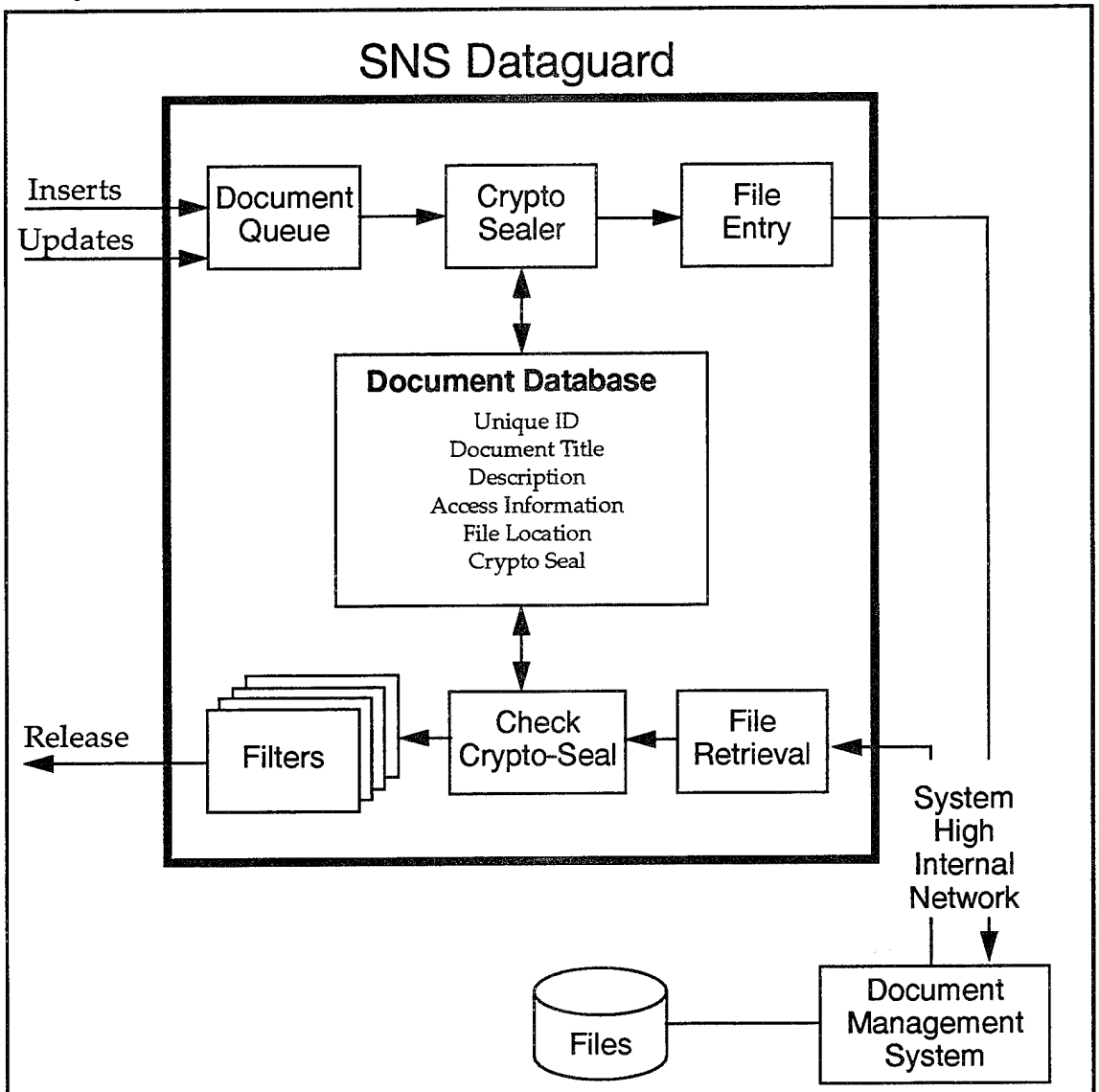


Figure 5: Providing an Archive of Tamper-detectable Documents

Two separate pipelines are used for implementing the SNS Dataguard: an Insert/Update pipeline in which new or updated documents are cryptographically sealed and passed to the CM system on the system high internal network for storage; and a Release pipeline for retrieving documents from the system high network, checking via the crypto seal that they have not been modified, and releasing them to the requestor after verifying that the requestor has permission to view the document. A document database, containing documents that are in the system and their associated crypto seals, is stored on the SNS system using LOCK DBMS.

ties or more security and integrity.

- Crypto-Seal at Remote Sites

  Authors submitting documents to the system could cryptographically seal the documents they produce before they are sent to the SNS Dataguard system. This eliminates the risk of the document being tampered with while it is being transported between the author and the SNS system. The cryptographic seals would be done using the author's own Fortezza card and Private Key. The Fortezza cards are part of the MISSI MOSAIC program and will be widely available. The MOSAIC program is addressing key distribution and key rollover issues. The SNS Dataguard will also add RSA based cryptography later.

  Upon receipt of a document that had been cryptographically sealed by an author, the SNS Dataguard system would verify that the crypto seal is accurate and was done using the author's Private Key. The document would then be entered into the system via the normal insertion pipeline. Recomputing the crypto seal using the SNS Private Key simplifies long-term key management issues and prevents the document from being directly entered into the system high network without going through the SNS Dataguard system.

  Allowing an author to cryptographically seal a document presents some additional issues that must be addressed. If an author loses their Fortezza card, they will need to report it immediately so that their key is entered on a Key Revocation List (KRL). The SNS Dataguard system would not accept any documents that have been sealed with a key on the KRL after the date the Fortezza card was lost. Also, the author's Public Key certificate, to be used to verify the seal on the document, could be stored with the crypto seal if it is needed in the future to check the author's crypto seal.

- Remote Access to Document Database

  Access could be provided to external users to browse an index to the document database and request documents. Requests for a list of documents in the database and for individual documents could be made via digitally signed email that the SNS Dataguard system could authenticate using the Fortezza card. This ensures that only legitimate requests are processed. It also prevents users from masquerading as other individuals to gain access to sensitive data. The SNS system could be configured so that all email to the system is routed to a mail daemon that is responsible for interpreting an email request and making the proper response. No email would be allowed to pass through the system.

  Release of documents could also be done by email, or possibly ftp. Once a document has been approved for release, it could be sent via email to the requestor or ftp'ed to the requestor's site. Before sending the document, it would be encrypted using the Fortezza card. Currently, this encryption uses

the Skipjack algorithm which is not approved for classified documents but could be used for unclassified but sensitive documents. Classified documents would still need to be sent via some other media. When Type 1 encryption is added to the MISSI system, however, classified documents could also be sent in this manner.

Another possible feature is to provide a World Wide Web interface to allow easy browsing of the document database.

- Flexible Access Control Policy

  Along with the multilevel security requirements, there are "need to know" restrictions placed on the release of documents to individuals outside the system high network.

  Access control information could be added to the databases as a tool for the release agent. This information allows the release agent to make better informed decisions and, by adding an enforcement mechanism, might eventually replace the release agent entirely. For example, if each document entry in the database had a list of organizations and individuals that were approved to see the document, an SNS trusted program could be written that only released a document to those on the document's approved list. Each document's approved list would be stored in the LOCK DBMS database and would be bound to the document's unique id by a crypto seal computed by the SNS Dataguard system. This prevents approval lists from being accidentally or maliciously altered but still allows them to be easily updated if needed.

- Multilevel Document Database Index

  In some cases the title of a document might be classified. As a result the titles cannot be stored in the unclassified database index for outside users to browse. Since the LOCK DBMS system is multilevel, a database index could be created at a higher level to store the classified document information. As a result only users cleared to the level of the document title would be able to see the document entry. In this way the existence of certain documents could be classified and protected. Users browsing the database index would only be given information on documents in the database at their security clearance or below. There would be no duplication of information because each document entry exists at one level. LOCK DBMS would provide a secure view of all the documents at or below the users clearance.

- Classes of Users

  Using the SNS type enforcement mechanism, it is possible to create user roles on the SNS that are mandatorily enforced. These roles could be used to enforce read/write versus read/only access to the LOCK DBMS database and to separate remote users into different classes that can view LOCK DBMS database information on different sets of documents and have different requirements checked before a document is released to them.

- Replicated SNS Systems

  In the initial configuration of the SNS Dataguard system, the SNS is a single point of failure. To increase the reliability of the system, a second SNS could be added that includes the same functionality as the first and that provides a backup system in case the original system goes down. The document database would be replicated on the second machine so that all document information is stored in two separate places. The replication could be handled either by using the distributed, replicated database functionality provided by ORACLE, or by building a custom program that keeps the databases on the separate systems synchronized.

The SNS Dataguard solves a significant problem that MLS document systems face: how to effectively and efficiently review documents for release from a system high network. The Dataguard is based on technology and products that currently exist and will be deployed within the next year. Once deployed, documents entered into the system would have associated crypto seals and later upgrades could be made to the system without affecting these documents or their crypto seals.

By incorporating a number of the features described in the previous section, a secure and flexible document insert and retrieval system can be developed. Figure 6 illustrates the long term vision. Users can insert documents, browse the database from an external network, request documents, have the request approved, and receive an encrypted copy in a manner of minutes.

5. Conclusion

LOCK DBMS provides the first opportunity to build highly assured, secure database applications. Based on the unique security features of SCC's SNS and the powerful DBMS features of Trusted Oracle V7, it offers an ideal platform for developing and investigating MLS DBMS applications. SCC has identified an initial set of interesting and important high assurance MLS applications and is beginning to develop them. SCC is also actively seeking other potential MLS DBMS applications and users interested in seeing them developed.

Author Info
haigh@sctc.com, obrien@sctc.com, thomsen@sctc.com
612-628-2700

Tom Haigh

Dr. Haigh is the Vice President and Director of Research at SCC. In this role he is responsible for developing SCC's Technology Acquisition Plan and for planning and implementing contract and IR&D programs that result in the acquisition of the identified technologies. Prior to assuming his current position, Dr. Haigh was Chief of the System Assurance Section at SCC. In that capacity, he was responsible for the program planning and technical direction of all SCC assurance efforts. Dr. Haigh has served on the program committees for the IEEE Symposium on Security and Privacy, the IEEE Workshop on the Foundations of Computer Security, and the Workshop of IFIP Working Group 11.3 on database security. He has written and presented many papers on computer security. One of his papers, "Extending the MLS Version of Noninterference for the Secure Ada Target", won outstanding paper award at the 1986 IEEE Symposium on Security and Privacy.

Dick O'Brien

Dr. O'Brien is a senior principal research scientist at SCC. Since joining SCC in 1987, he has been primarily involved with the design and analysis of trusted systems. Most recently, he was the principal investigator on the LOCK DBMS program that resulted in the development of a high assurance MLS DBMS based on a COTS product. On the LOCK project, he worked with both the design team, writing module level design specifications, and the assurance team, leading the Class A1 assurance effort for the LOCK kernel extensions and the specification to code correspondence. Prior to joining SCC, Dr. O'Brien was an associate professor teaching mathematics and computer science for nine years. He has published several papers in the areas of computer security, mathematical analysis and graph theory.

Dan Thomsen

Mr. Thomsen is a principal research scientist at SCC involved in trusted system development and trusted DBMS research. He has been heavily involved in the secure database efforts at SCC and was the lead technical engineer on the LOCK DBMS program with responsibility for porting Trusted Oracle to the LOCK platform. He has also worked in many areas of TCB development under the LOCK program. Mr. Thomsen has a Master's degree in Computer Science, specializing in computer security. He has published several research papers discussing Trusted DBMS issues and computer integrity policies.

USING A SECURE GUARD WITH DATA REPLICATION

RICK UHRIG, SYBASE, INC.

1. OVERVIEW

This paper describes a contractor developed prototype for replicating data between systems at different classification levels. The prototype was developed as a proof of concept demonstration for the Global Transportation Network (GTN). The work was performed by Systems Research and Applications Corporation (SRA), with support from HFSI and Sybase.

The approach combines an all COTS replication solution with a contractor developed security guard. The guard assures that information only flows one way — from low to high. It prevents high information from leaking across to the low system.

The data replication technology satisfies certain key requirements:

- It is an all COTS solution, assuring system viability and reliability
- It is transaction-based, assuring the same high level of data integrity at the replicated system as in the originating system.
- It occurs in near-real time, providing up-to-data information in the replicated system.
- It is automatic, eliminating the need for a man in the loop.

The secure guard is also based on a foundation of existing COTS technology — HFSI's XTS-300 has been evaluated by the NCSC and appears on the evaluated products list. For the prototype, contractor developed software was added to support replication. It allows data to be replicated from the low to high environments, while assuring that no data flows in the wrong direction. High data is not allowed to spill over into the low system.

To enable replication, the guard must fit with the other components of the architecture. For the RDBMS engine, it must mimic the data replication mechanism; and for the data replication mechanism, it must mimic the RDBMS engine. Further, to assure that it is truly secure, the guard must depend on only the evaluated security mechanisms of the hardware and operating system. This paper shows the specifics of how this is accomplished in the prototype.

Finally, this paper discusses the tradeoffs between functionality and security involved with closing the covert channels, and the solution that was chosen for the prototype.

2. THE GTN ENVIRONMENT

When completed, the Global Transportation Network will comprise 4 sites that are linked in pairs, each site within the pair serving as either a primary or alternate. Each site will be divided into two system-high security partitions — one at Sensitive Unclassified, the other at TOP SECRET —

with additional interfaces to other SECRET systems. All customer systems, user terminals, and workstation applications run system-high and only connect to either the Unclassified or TOP SECRET partitions.

A one-way security guard will facilitate secure data transfer from the Unclassified partition to the TOP SECRET partition. This is the guard that has been prototyped and is described in this paper. With this guard, data received from Unclassified sources is used to automatically populate databases within both the Unclassified and TOP SECRET partitions. The Unclassified database is updated first, then the changes are replicated through the secure guard to the TOP SECRET database. This is illustrated in Figure 1.

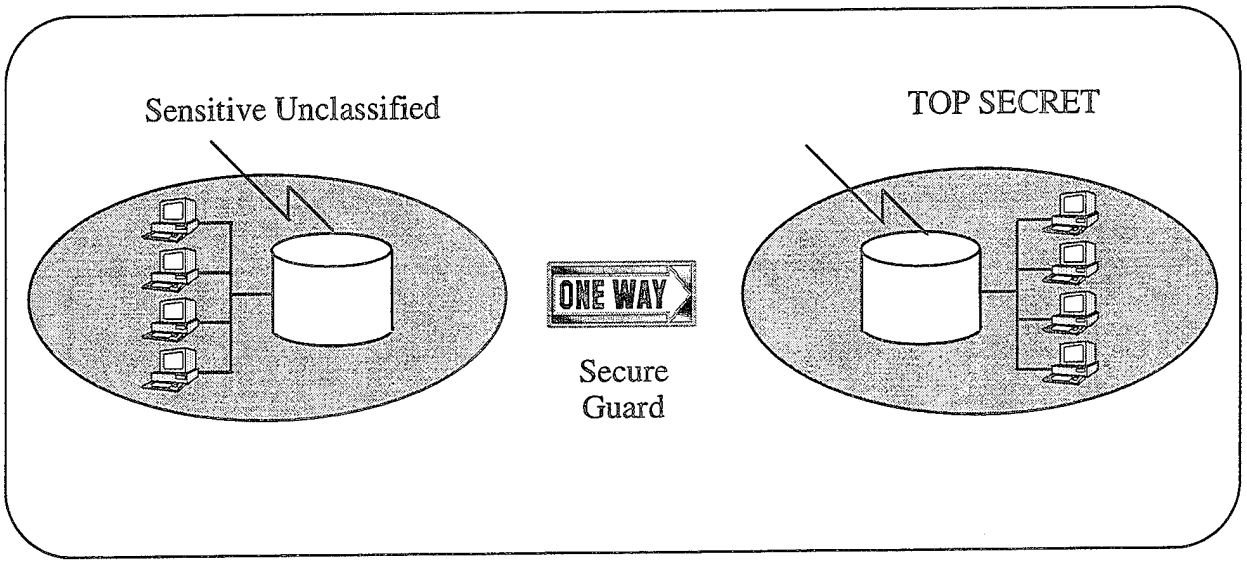


Figure 1. Each GTN site includes Sensitive Unclassified and TOP SECRET partitions. Unclassified Data is used to populate servers in the Unclassified partition. This information is then replicated through the one-way guard to the TOP SECRET partition.

Though not pertinent to the effort described here, it is worth noting that other one-way security guards are planned to allow data transfer from the Unclassified partition to the other SECRET systems, and from these SECRET systems to the TOP SECRET partition.

3. DATA REPLICATION

Data Replication is a key technology for GTN. This section provides background on data replication.

The Origins of Data Replication

In the last few years, data replication has emerged as the technology of choice for distributed data management. Earlier attempts with *two phase commit* (2PC) largely failed, primarily because of performance bottlenecks and system availability limitations that are built-in to 2PC. 2PC follows

a *tight synchronization* model, where all copies of a data element are identical, or recoverable to an identical state. As one might imagine, tight consistency introduces certain inefficiencies and overhead that directly result in the observed bottlenecks and availability limitations. Customers that attempted to implement distributed data management systems using 2PC quickly became dissatisfied. Replication is the innovation that was born of that dissatisfaction.

Data Replication follows a *loose synchronization* model. Unlike tight synchronization, all locations are not required to update a data item at the same time. Rather, one site (the primary or originating site) completes its update, and then forwards that change to other sites (the replicate sites) for action. Although the data elements are not kept identical, they are nearly so. The latency period between the update at the primary and the corresponding update at the replicates is only a matter of moments (usually measured in seconds, rarely in minutes).

In practice, the loose synchronization model meets the business requirements of almost all customers and avoids the performance bottlenecks and availability limitations of tight synchronization.

SYBASE Replication

Sybase provides data replication through the *SYBASE Replication Server*. When changes occur in the database, a *Replication Agent* forwards those changes to the Replication Server. The Replication Server then checks its list of *subscriptions* to determine which changes need to be replicated to other databases and forwards those as appropriate. This is illustrated in Figure 2.

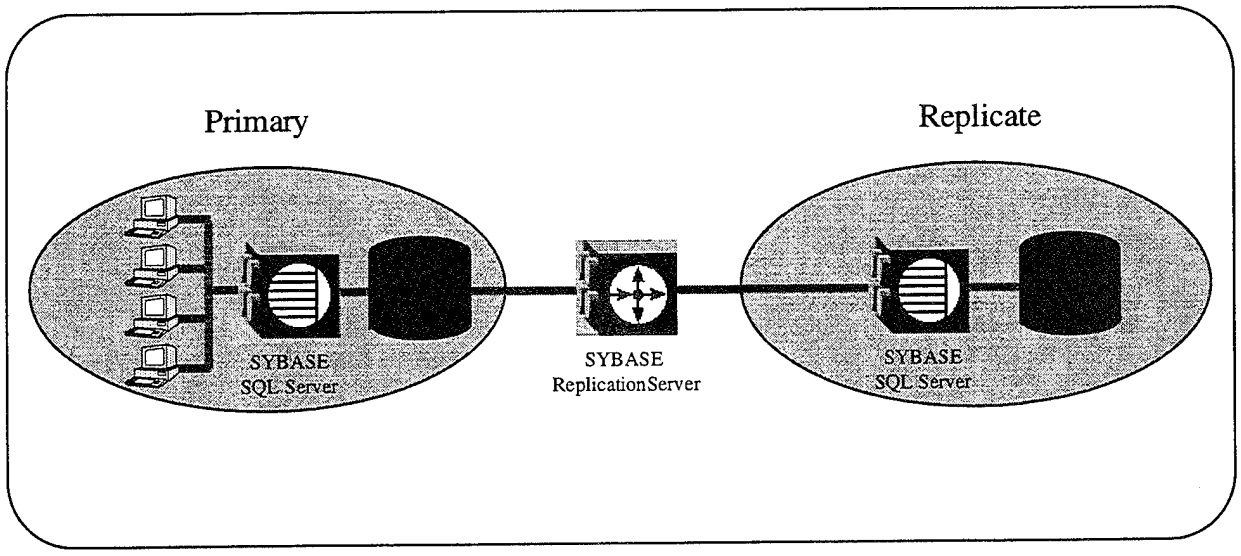


Figure 2. A simple example of SYBASE Replication. User applications update the primary database through the RDBMS (SQL Server). A Replication Agent (not shown) forwards the update to the Replication Server, which then determines if that update has been subscribed to. If it has, Replication Server passes the update along to the replicate SQL Server which then updates the replicate database. Note that this example was kept simple for clarity. More generally, Replication server can forward updates to multiple replicates, and databases can simultaneously be primary for some data items and replicates for others.

Benefits of Replication

COTS Based. Generally speaking, commercial products have much greater viability and reliability than "home grown" or "stove pipe" solutions. Virtually no one considers writing their own operating system or DBMS from scratch. With good commercial products available, it doesn't make sense. For the same reasons, no one should consider writing their own distributed data management software.

Automated Operations. Some distributed data management solutions require a human in the loop (e.g. to perform a tape transfer.) Once configured, Replication Server fully automates the distributed data function and eliminates the need for human intervention.

High Availability. Replication Server's loose synchronization model assures high availability. Even if one component is unavailable, other components will be able to commit their updates. For example, a primary site will be able to proceed with its updates even though a replicate site may be off-line.

Reliable Delivery. The Replication Server is designed with the realization that, in the real world, components do fail. Replication Server uses *store & forward* queues to assure that updates are not lost. When a "downstream" component is unavailable, the update is stored in the queue until that component becomes available and can accept the update.

Data Integrity. The transaction is the basic unit of data integrity with a DBMS. It is also the basic unit of replication within the Replication Server (i.e. Replication Server replicates complete transactions rather than individual data items), assuring that complete transactions arrive at the replicate sites. This assures the data integrity at the replicate sites

Timely Delivery. Replication Server is event driven. When an update transaction occurs at the primary, the update is forwarded to the Replication Server, which then sends it to the appropriate replicates. The net result is that replicate sites receive the update transaction in near real time – usually in a matter of moments.

Efficient. Replication Server is highly efficient since only the changes need to be replicated. Consider for example a 100 MB database where only 1 MB of data changes in a given period of time. Only that changed 1 MB needs to be replicated. In fact, less may be replicated, depending on what has been subscribed to.

With other approaches to distributed data management it is often difficult to determine what data has changed. With those approaches all data must be replicated, even if most of it hasn't changed. In the given example, this would amount to 9,900% overhead!

Open. Replication Server supports replicating data into and out of non-Sybase databases, including DB2 and Oracle.

4. GUARD TECHNOLOGY

The Replication Server can be configured to replicate in only one direction. In fact, for this prototype, the subscriptions are set up so that data only flows from the Unclassified to TOP SECRET partitions. However, without additional assurances, such a system would not be accredited for operational use.

Security guards provide that additional assurance. They act as traffic cops – enforcing restrictions on data flow, and more importantly, providing a high level of assurance that those restrictions are correctly enforced.

HFSI's XTS-300 was chosen for the security guard. The XTS-300 combines Intel 486 based hardware, PC commodity peripherals, the STOP 4.0 trusted operating system and UNIX user interfaces into a system meeting NCSC B3 requirements. Connection of the XTS-300 to networks is provided by the Secure Communications Subsystem, which is based on the Intel 386. The Secure Communications Subsystem off-loads the XTS-300 of the burden lower layer network protocol processing.

For the GTN prototype the XTS-300 was configured with two Secure Communications Subsystems — one each for the Unclassified and TOP SECRET partitions. This is illustrated in Figure 3.

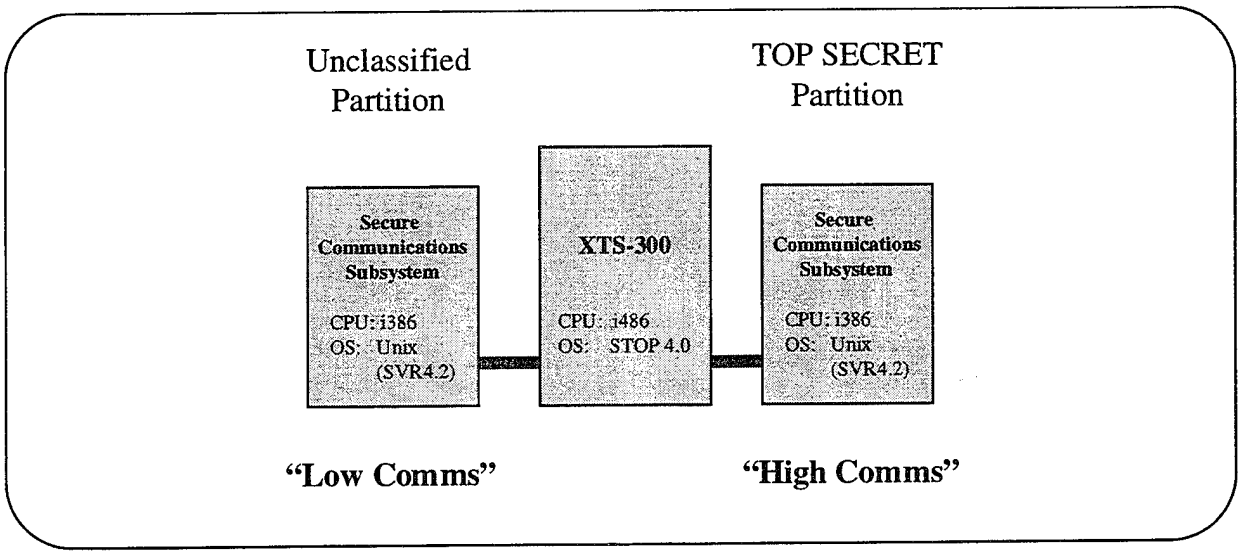


Figure 3. The XTS-300 is configured with two Secure Communications Subsystems, one for the Unclassified partition, and one for the TOP SECRET partition.

5. PUTTING IT ALL TOGETHER

Integrating SYBASE's Replication Server with the XTS-300 turned out to be straight-forward. The trick was to translate the database network protocol that Replication Server understood into

something that the XTS-300 could work with. The XTS-300 already had a trusted *file transfer protocol* (FTP), so the decision was made to unwind the database network protocol into a flat file, hand that off to the XTS-300 to transfer up to the TOP SECRET security partition, and then translate the contents of the flat file back into the database networking protocol.

Several software components were developed to make this happen. *Guard Sender* accepts the output stream from the Replication Server, translates the stream into SQL statements, and deposits these in a flat file in a known location.

Developed software on the Low Comms processor polls the known location for new files. When it discovers a new file, it initiates a file transfer up to the TOP SECRET partition. The XTS-300 performs the file transfer, then with developed software on the High Comms processor, places the file where it can be found by the *Guard Catcher.*

Guard Catcher polls for the appearance of new files. Upon finding a new file, the Guard Catcher translates the enclosed SQL statements back to the database network protocol, reversing the action of Guard Sender. The output of Guard Catcher is forwarded directly to the TOP SECRET SQL Server. This is illustrated in Figure 4.

Both Guard Sender and Guard Catcher are fairly straight-forward applications. They are built using SYBASE Open Client and Sybase Open Server, Sybase's APIs for the database network protocol. This made tapping into the protocol easy. In fact, using this API, Guard Sender appears just like a SQL Server (the Replication Server can't tell the difference) and Guard Catcher appears just like a Replication Server (the SQL Server can't tell the difference).

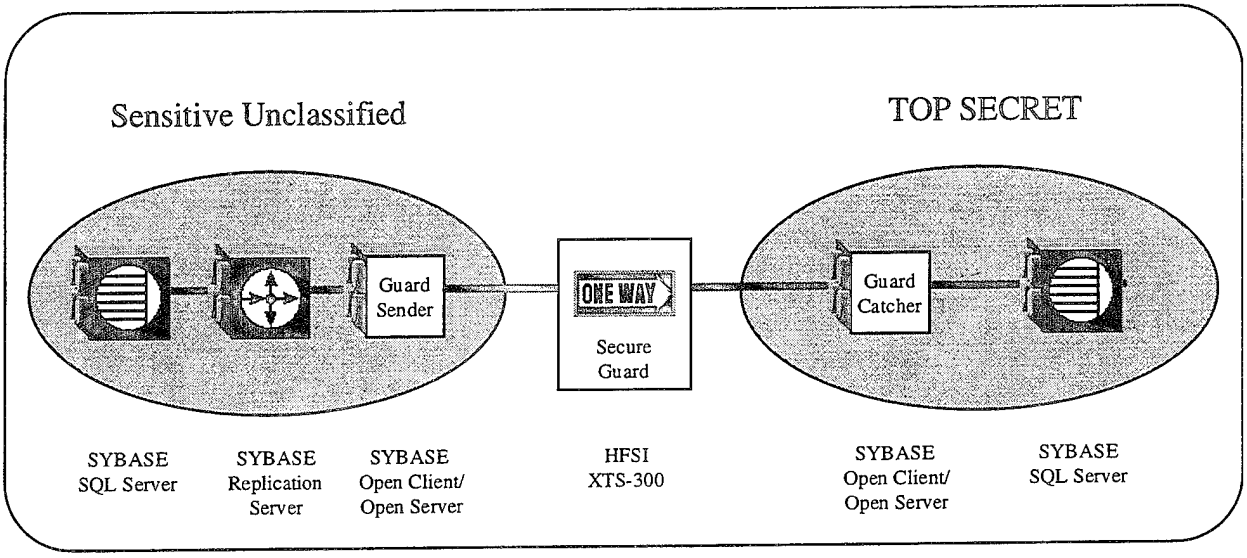


Figure 4. The architected solution for replicating with a secure guard. The *Guard Sender* unwinds the database network protocol into a flat file format. The XTS-300 detects the arrival of the flat file on the low side, and performs a trusted FTP to move the file from the Sensitive Unclassified partition to the TOP SECRET partition. *The Guard Catcher* detects the arrival of the flat file on the high side and converts it back into the database network protocol so that the replication can proceed.

6. COVERT CHANNEL CONSIDERATIONS

It is important to note that, for data replication is to be reliable, some control information must be passed back to the low environment. The Replication Server must get, via the guard, the following types of feedback:

- Acknowledgments for successful replication updates
- Error codes for unsuccessful replication updates
- Certain information needed for restart and recovery operations.

This of course is in conflict with the guard's mission, which is meant to assure that information flows in only one direction. However, reliable operation of any replication scheme requires that some control information flow back downhill. This is the basic security tradeoff — how much control information can be passed back downhill without compromising the security and accreditability of the system.

For this prototype the decision was made to close down the covert channel. The XTS-300 automatically acknowledges all replications, whether they actually get all the way to the database or not. This approach was chosen because it mitigated accreditation risk.

A future option is to enhance the prototype so that it can pass back error codes for unsuccessful replication updates. The bandwidth of the covert channel can be minimized by passing back only a single generic error code rather than the more specific and detailed error codes that are available.

Restart and recovery operations that cross classification level will probably always require a human in the loop. There appears to be little if any benefit in enhancing the guard to pass back this control information.

7. CONCLUSION

This effort was succeeded. It accomplished its goal of integrating COTS data replication technology with a secure guard.

The Author

Rick Uhrig is a Principal System Consultant at Sybase, Inc, 6550 Rock Spring Drive, Suite 800, Bethesda, MD 20817. Rick has been serving time in Sybase's Federal Business Development organization for two and one half years, where his technical responsibilities include multilevel security, database interoperability, and distributed architectures.

Rick was previously employed by McDonnell Douglas for nine years, where his work included systems integration and multilevel security. Rick has master's degrees from the University of Wisconsin in Mathematics and Computer Science.

Rick can be reached at (301) 897-1739 and FAX (301) 897-1601. His E-mail address is richard.uhrig@sybase.com

THE IMPACT OF DECLASSIFYING NATIONAL SECURITY INFORMATION ON DATA MANAGEMENT

Hernan I. Otano, Richard S. Carson & Associates, Inc.

Introduction

On April 17, 1995, an Executive Order (EO) number 12958 titled "Classified National Security Information" was signed by the President. In Section 3.4 of the Executive Order it specifically states: "within 5 years from the date of this order, all classified information contained in records that (1) are more than 25 years old, and (2) have been determined to have permanent historical value under title 44, United States Code, **shall be automatically declassified whether or not the records have been reviewed."** This means that every single page must be read and processed in compliance with the EO and to avoid any sanctions as delineated in Section 5.7. (**"Officers and employees of the United States Government, and its contractors, licensees, certificate holders, and grantees shall be subject to appropriate sanctions if they knowingly, willfully, or negligently ..."**) Due to the large quantities of information within the government it is essential that the Federal agencies understand the EO and how it affects their agency and what is required in order to comply with the EO. There are many commercial off-the-shelf (COTS) products on the market today that can be used or integrated as a technical solution that may be implemented to help government agencies manage their information. However, if the technical solution implemented is not well thoughout, the government agency can be greatly impacted. This paper will specifically address a technical solution that can serve as a model that complies with Section 3.4, "Automatic Declassification."

1. Background

A particular federal government agency has over eighty million faces of paper and an uncounted number of three-dimensional objects, motion picture and still film, microfiche, microfilm, aperture cards computer tapes, video, and audio in various formats. During the first year of this project, the main focus is on the paper documents which it is the bulk of the materials. The task is to develop a technical solution that would not only handle the large quantity of information and different types of information but would also comply with the EO. The primary concern is the time consumed in each document handling operation. After a sample survey of the archives, it was estimated more than one half of the eighty million faces will be exempted leaving forty million faces to be processed. Based on past experience, the estimated number for handling paper documents is always low.

The task was to develop a technical solution that would not only handle the large quantity of information and different types of information but would also comply with the EO.

2. Technical Solution

Fortunately, with the present state-of-the-art imaging systems makes it possible to tackle a project of this size using a COTS product. For this particular project, a complete scaleable, working system has been built based on Windows 3.11 and Windows NT operating system. This system consists of the following basic modules all connected and integrated on a secure network:

- Image Capture Stations
- Image Storage Servers
- Review and Redaction Stations
- Integrated Work Tracking and Scheduling Subsystems

Image Capture Stations consist of two basic types of scanners, (1) loose sheets automatic feed straight path types and (2) flat bed manual feed for fragile materials. The control menu screen presentation is identical for both types. The image is first stored on the workstation local hard disk and after quality control, the images are downloaded to the servers. For primary index is the referencing to the document container (in this case eight (8) alphanumeric characters) is entered creating a directory; and each set of documents, as they were originally, stored in the container are entered as subdirectories.

Image Storage Servers are magnetic in a fault tolerant scaleable Redundant Array of Inexpensive Disks (RAID) type 5 configuration. The anticipated storage needs for this project are three (3) terabytes.

Review and Redaction Stations are the key to the operation. This station has the capability to annotate the image in free form creating search terms that couple with a full text search engine that retrieves the image. The choice for a full text storage and retrieval engine was made because field and keywords will not effectively work with unknown material content. In addition, having a full face or partial face OCR capability may also help in the retrieval. However, because the age, condition and type of documents, it is expected that OCR will have minor use.

Redaction of the image is performed in two steps. First, by using transparent colors the proposed image modifications are shown. Second the final authority(s) approve or modify the recommendations and create the final redacted image version by converting the transparent color to solid black creating a complete new image.

As a final release product there is one redacted image for each face of a document with corresponding redacted text either as a keyboard entry (annotations) or redacted OCR if it is available. In addition for internal use only, all original and redacted images and corresponding text will be kept in off-line storage.

Integrated Workflow and Scheduling Subsystems. The program required several teams of diverse specialized human talent. It is not expected that a person can effectively read and assimilate information for more than four to five hours, more than one shift per day are employed requiring the use of a personnel scheduling mechanism. In addition and coupled with the

scheduling subsystem, means to monitor and control the work in progress and completed it is necessary (commonly known as workflow). This is the only COTS product which is under evaluation because there are over 50 choices.

3. Criteria

When considering a technical solution for document and image management, one must establish selection criteria. The main selection criteria is that the software must be reconfigurable by the user without the need to enter codes and preferable by drag and drop operations. We are well aware the rapid changes in technology and for these reasons we must adopt well establish standards and avoiding proprietary products. For images we use Government standard CITT Group 4 no loss compression and for text ASCII generated either by keyboard or by the OCR process.

Be aware of manufacturer specifications in selecting hardware and software. The best way to determine hardware and software compatibility is to have an on-site evaluation with all the components and the test should be made with actual materials.

4. Conclusion

As mandated by the Executive Order, government agencies must comply. There are many COTS products that can be integrated to serve as a complete technical solution. Without a sound and well thoughout technical solution an organization can be severely impacted. Attached is the complete text of the Executive Order with portions bolded that specifically address Sections 3.4 and 5.7.

By the time the Colloquium is in sessions, I will be able to present more information about this program such as statistics, timings and other findings.

Biography and Related Experience

Hernan Otano, a Senior Associate with Richard S. Carson & Associates, Inc., has over twenty years experience in the areas of document image, storage and retrieval, digital optical storage, data and image compression, full text search and retrieval, optical character recognition, local and wide area networks, multimedia and interactive technologies. Mr. Otano is currently working with DoD and other Federal Agencies implementing storage and retrieval of document images, full text search and management capabilities using commercial software technology.

In 1987 Mr. Otano founded a corporation dedicated to developing document management systems using full text indexing with automatic image retrieval and optical storage. Some of the systems include: Executive Office of the President (Presidential correspondence), Spanish Supreme Court, historical documents, dating from the 14th Century related with the Spanish discover and Colonization of the Americas, International Trade Commission, pilot project for the Federal Reserve Board, U.S. Army, Reserve Component Automation System among others.

In 1978, as a federal employee at the Smithsonian National Air and Space Museum, Mr. Otano designed and implemented a historical photographic archival project where materials were recorded in Video Disc under different video standards for worldwide distribution. In 1984 designed and implemented a Historical Document Storage and Retrieval system based on optical digital technology and full text indexing. The first operational system in the U.S. using 12 inch disks with capacity of 3.2 Gigabytes each.

Hernan Otano
Senior Associate
Richard S. Carson & Associates, Inc.
4330 East-West Highway, Suite 304
Bethesda, Maryland 20814

phone: (301)656-4565
fax: (301)656-4806
internet: otano@carsoninc.com
or
hotano@ra.osd.mil

EXECUTIVE ORDER 12958 April 17, 1995

CLASSIFIED NATIONAL SECURITY INFORMATION

This order prescribes a uniform system for classifying, safeguarding, and declassifying national security information. Our democratic principles require that the American people be informed of the activities of their Government. Also, our Nation's progress depends on the free flow of information. Nevertheless, throughout our history, the national interest has required that certain information be maintained in confidence in order to protect our citizens, our democratic institutions, and our participation within the community of nations. Protecting information critical to our Nation's security remains a priority. In recent years, however, dramatic changes have altered, although not eliminated, the national security threats that we confront. These changes provide a greater opportunity to emphasize our commitment to open Government.

NOW, THEREFORE, by the authority vested in me as President by the Constitution and the laws
of the United States of America, it is hereby ordered as follows:

PART 1-- ORIGINAL CLASSIFICATION

Section 1.1. Definitions. For purposes of this order:

(a) "National security" means the national defense or foreign relations of the United States.

(b) "Information" means any knowledge that can be communicated or documentary material, regardless of its physical form or characteristics, that is owned by, produced by or for, or is under the control of the United States Government. "Control" means the authority of the agency that originates information, or its successor in function, to regulate access to the information.

(c) "Classified national security information" (hereafter "classified information") means information that has been determined pursuant to this order or any predecessor order to require protection against unauthorized disclosure and is marked to indicate its classified status when in documentary form.

(d) "Foreign Government Information" means:

(1) information provided to the United States Government by a foreign government or governments, an international organization of governments, or any element thereof, with the expectation that the information, the source of the information, or both, are to be held in confidence;

(2) information produced by the United States pursuant to or as a result of a joint arrangement with a foreign government or governments, or an international organization of governments, or any element thereof, requiring that the information, the arrangement, or both, are to be held in confidence; or

(3) information received and treated as "Foreign Government Information" under the terms of a predecessor order.

(e) "Classification" means the act or process by which information is determined to be classified information.

(f) "Original classification" means an initial determination that information requires, in the interest of national security, protection against unauthorized disclosure.

(g) "Original classification authority" means an individual authorized in writing, either by the President, or by agency heads or other officials designated by the President, to classify information in the first instance.

(h) "Unauthorized disclosure" means a communication or physical transfer of classified information to an unauthorized recipient.

(i) "Agency" means any "Executive agency," as defined in 5 U.S.C. 105, and any other entity within the executive branch that comes into the possession of classified information.

(j) "Senior agency official" means the official designated by the agency head under section 5.6(c) of this order to direct and administer the agency's program under which information is classified, safeguarded, and declassified.

(k) "Confidential source" means any individual or organization that has provided, or that may reasonably be expected to provide, information to the United States on matters pertaining to the national security with the expectation that the information or relationship, or both, are to be held in confidence.

(l) "Damage to the national security" means harm to the national defense or foreign relations of the United States from the unauthorized disclosure of information, to include the sensitivity, value, and utility of that information.

Sec. 1.2. Classification Standards. (a) Information may be originally classified under the terms of this order only if all of the following conditions are met:

(1) an original classification authority is classifying the information;

(2) the information is owned by, produced by or for, or is under the control of the United States Government;

(3) the information falls within one or more of the categories of information listed in section 1.5 of this order; and

(4) the original classification authority determines that the unauthorized disclosure of the information reasonably could be expected to result in damage to the national security and the original classification authority is able to identify or describe the damage.

(b) If there is significant doubt about the need to classify information, it shall not be classified. This provision does not:

(1) amplify or modify the substantive criteria or procedures for classification; or

(2) create any substantive or procedural rights subject to judicial review.

(c) Classified information shall not be declassified automatically as a result of any anauthorized disclosure of identical or similar information.

Sec. 1.3. Classification Levels. (a) Information may be classified at one of the following three levels:

(1) "Top Secret" shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause exceptionally grave damage to the national security that the original classification authority is able to identify or describe.

(2) "Secret" shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause serious damage to the national security that the original classification authority is able to identify or describe.

(3) "Confidential" shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause damage to the national security that the original classification authority is able to identify or describe.

(b) Except as otherwise provided by statute, no other terms shall be used to identify United States classified information.

(c) If there is significant doubt about the appropriate level of classification, it shall be classified at the lower level.

Sec. 1.4. Classification Authority. (a) The authority to classify information originally may be exercised only by:

(1) the President;

(2) agency heads and officials designated by the President in the Federal Register; or

(3) United States Government officials delegated this authority pursuant to paragraph (c), below.

(b) Officials authorized to classify information at a specified level are also authorized to classify information at a lower level.

(c) Delegation of original classification authority.

(1) Delegations of original classification authority shall be limited to the minimum required to administer this order. Agency heads are responsible for ensuring that designated subordinate officials have a demonstrable and continuing need to exercise this authority.

(2) "Top Secret" original classification authority may be delegated only by the President or by an agency head or official designated pursuant to paragraph (a)(2), above.

(3) "Secret" or "Confidential" original classification authority may be delegated only by the President; an agency head or official designated pursuant to paragraph (a)(2), above; or the senior agency official, provided that official has been delegated "Top Secret" original classification authority by the agency head.

(4) Each delegation of original classification authority shall be in writing and the authority shall not be redelegated except as provided in this order. Each delegation shall identify the official by name or position title.

(d) Original classification authorities must receive training in original classification as provided in this order and its implementing directives.

(e) Exceptional cases. When an employee, contractor, licensee, certificate holder, or grantee of an agency that does not have original classification authority originates information believed by that person to require classification, the information shall be protected in a manner consistent with this order and its implementing directives. The information shall be transmitted promptly as provided under this order or its implementing directives to the agency that has appropriate subject matter interest and classification authority with respect to this information. That agency shall decide within 30 days whether to classify this information. If it is not clear which agency has classification responsibility for this information, it shall be sent to the Director of the Information Security Oversight Office. The Director shall determine the agency having primary

subject matter interest and forward the information, with appropriate recommendations, to that agency for a classification determination.

Sec. 1.5. Classification Categories.
Information may not be considered for classification unless it concerns:

(a) military plans, weapons systems, or operations;

(b) foreign government information;

(c) intelligence activities (including special activities), intelligence sources or methods, or cryptology;

(d) foreign relations or foreign activities of the United States, including confidential sources;

(e) scientific, technological, or economic matters relating to the national security;

(f) United States Government programs for safeguarding nuclear materials or facilities; or

(g) vulnerabilities or capabilities of systems, installations, projects or plans relating to the national security.

Sec. 1.6. Duration of Classification. (a) At the time of original classification, the original classification authority shall attempt to establish a specific date or event for declassification based upon the duration of the national security sensitivity of the information. The date or event shall not exceed the time frame in paragraph (b), below.

(b) If the original classification authority cannot determine an earlier specific date or event for declassification, information shall be marked for declassification 10 years from the date of the original decision, except as provided in paragraph (d), below.

(c) An original classification authority may extend the duration of classification or reclassify specific information for successive periods not to exceed 10 years at a time if such action is consistent with the standards and procedures established under this order. This provision does not apply to information contained in records that are more than 25 years old and have been determined to have permanent historical value under title 44, United States Code.

(d) At the time of original classification, the original classification authority may exempt from declassification within 10 years specific information, the unauthorized disclosure of which could reasonably be expected to cause damage to the national security for a period greater than that provided in paragraph (b), above, and the release of which could reasonably be expected to:

(1) reveal an intelligence source, method, or activity, or a cryptologic system or activity;

(2) reveal information that would assist in the development or use of weapons of mass destruction;

(3) reveal information that would impair the development or use of technology within a United States weapons system;

(4) reveal United States military plans, or national security emergency preparedness plans;

(5) reveal foreign government information;

(6) damage relations between the United States and a foreign government, reveal a confidential source, or seriously undermine diplomatic activities that are reasonably expected to be ongoing for a period greater than that provided in paragraph (b),above;

(7) impair the ability of responsible United States Government officials to protect the President, the Vice President, and other individuals for whom protection services, in the interest of national security, are authorized; or

(8) violate a statute, treaty, or international agreement.

(e) Information marked for an indefinite duration of classification under predecessor orders, for example, "Originating Agency's Determination Required," or information classified underpredecessor orders that contains no declassification instructions shall be declassified in accordance with part 3 of this order.

Sec. 1.7. Identification and Markings. (a) At the time of original classification, the following shall appear on the face of each classified document, or shall be applied to other classified media in an appropriate manner:

(1) one of the three classification levels defined in section 1.3 of this order;

(2) the identity, by name or personal identifier and position, of the original classification authority;

(3) the agency and office of origin, if not otherwise evident;

(4) declassification instructions, which shall indicate one of the following:

(A) the date or event for declassification, as prescribed in section 1.6(a) or section 1.6(c); or

(B) the date that is 10 years from the date of original classification, as prescribed in section 1.6(b); or

(C) the exemption category from declassification, as prescribed in section 1.6(d); and

(5) a concise reason for classification which, at a minimum, cites the applicable classification categories in section 1.5 of this order.

(b) Specific information contained in paragraph (a), above, may be excluded if it would reveal additional classified information.

(c) Each classified document shall, by marking or other means, indicate which portions are classified, with the applicable classification level, which portions are exempt from declassification under section 1.6(d) of this order, and which portions are unclassified.In accordance with standards prescribed in directives issued under this order, the Director of the Information Security Oversight Office may grant waivers of this requirement for specified classes of documents or information. The Director shall revoke any waiver upon a finding of abuse.

(d) Markings implementing the provisions of this order, including abbreviations and requirements to safeguard classified working papers, shall conform to the standards prescribed in implementing directives issued pursuant to this order.

(e) Foreign government information shall retain its original classification markings or shall be assigned a U.S. classification that provides a degree of protection at least equivalent to that required by the entity that furnished the information.

(f) Information assigned a level of classification under this or predecessor orders shall be considered as classified at that level of classification despite the omission of other required markings. Whenever such information is used in the derivative classification process or is reviewed for possible declassification, holders of such information shall coordinate with an appropriate classification authority for the application of omitted markings.

(g) The classification authority shall, whenever practicable, use a classified addendum whenever classified information constitutes a small portion of an otherwise unclassified document.

Sec. 1.8. Classification Prohibitions and Limitations. (a) In no case shall information be classified in order to:

(1) conceal violations of law, inefficiency, or administrative error;

(2) prevent embarrassment to a person, organization, or agency;

(3) restrain competition; or

(4) prevent or delay the release of information that does not require protection in the interest of national security.

(b) Basic scientific research information not clearly related to the national security may not be classified.

(c) Information may not be reclassified after it has been declassified and released to the public under proper authority.

(d) Information that has not previously been disclosed to the public under proper authority may be classified or reclassified after an agency has received a request for it under the Freedom of Information Act (5 U.S.C. 552) or the Privacy Act of 1974 (5 U.S.C. 552a), or the mandatory review provisions of section 3.6 of this order only if such classification meets the requirements of this order and is accomplished on a document-by-document basis with the personal participation or under the direction of the agency head, the deputy agency head, or the senior agency official designated under section 5.6 of this order. This provision does not apply to classified information contained in records that are more than 25 years old and have been determined to have permanent historical value under title 44, United States Code.

(e) Compilations of items of information which are individually unclassified may be classified if the compiled information reveals an additional association or relationship that:

(1) meets the standards for classification under this order; and

(2) is not otherwise revealed in the individual items of information.

As used in this order, "compilation" means an aggregation of pre-existing unclassified items of information.

Sec. 1.9. Classification Challenges. (a) Authorized holders of information who, in good faith, believe that its classification status is improper are encouraged and expected to challenge the classification status of the information in accordance with agency procedures established under paragraph (b), below.

(b) In accordance with implementing directives issued pursuant to this order, an agency head or senior agency official shall establish procedures under which authorized holders of information are encouraged and expected to challenge the classification of information that they believe is improperly classified or unclassified. These procedures shall assure that:

(1) individuals are not subject to retribution for bringing such actions;

(2) an opportunity is provided for review by an impartial official or panel; and

(3) individuals are advised of their right to appeal agency decisions to the Interagency Security Classification Appeals Panel established by section 5.4 of this order.

PART 2 DERIVATIVE CLASSIFICATION

Sec. 2.1. Definitions. For purposes of this order: (a) "Derivative classification" means the incorporating, paraphrasing, restating or generating in new form information that is already classified, and marking the newly developed material consistent with the classification markings that apply to the source information. Derivative classification includes the classification of information based on classification guidance. The duplication or reproduction of existing classified information is not derivative classification.

(b) "Classification guidance" means any instruction or source that prescribes the classification of specific information.

(c) "Classification guide" means a documentary form of classification guidance issued by an original classification authority that identifies the elements of information regarding a specific subject that must be classified and establishes the level and duration of classification for each such element.

(d) "Source document" means an existing document that contains classified information that is incorporated, paraphrased, restated, or generated in new form into a new document.

(e) "Multiple sources" means two or more source documents, classification guides, or a combination of both.

Sec. 2.2. Use of Derivative Classification. (a) Persons who only reproduce, extract, or summarize classified information, or who only apply classification markings derived from source material or as directed by a classification guide, need not possess original classification authority.

(b) Persons who apply derivative classification markings shall:

(1) observe and respect original classification decisions; and

(2) carry forward to any newly created documents the pertinent classificationmarkings. For information derivatively classified based on multiple sources, the derivative classifier shall carry forward:

(A) the date or event declassification that corresponds to the longest period of classification among the sources; and

(B) a listing of these sources on or attached to the official file or record copy.

Sec. 2.3. Classification Guides. (a) Agencies with original classification authority shall prepare classification guides to facilitate the proper and uniform derivative classification of information. These guides shall conform to standards contained in directives issued under this order.

(b) Each guide shall be approved personally and in writing by an official who:

(1) has program or supervisory responsibility over the information or is the senior agency official; and

(2) is authorized to classify information originally at the highest level of classification prescribed in the guide.

(c) Agencies shall establish procedures to assure that classification guides are reviewed and updated as provided in directives issued under this order.

PART 3 DECLASSIFICATION AND DOWNGRADING

Sec. 3.1. Definitions. For purposes of this order: (a) "Declassification" means the authorized change in the status of information from classified information to unclassified information.

(b) "Automatic declassification" means the declassification of information based solely upon:

(1) the occurrence of a specific date or event as determined by the original classification authority; or

(2) the expiration of a maximum time frame for duration of classification establishedunder this order.

(c) "Declassification authority" means:

(1) the official who authorized the original classification, if that official is still serving in the same position;

(2) the originator's current successor in function;

(3) a supervisory official of either; or

(4) officials delegated declassification authority in writing by the agency head or the senior agency official.

(d) "Mandatory declassification review" means the review for declassification of classified information in response to a request for declassification that meets the requirements under section 3.6 of this order.

(e) "Systematic declassification review" means the review for declassification of classified information contained in records that have been determined by the Archivist of the United States ("Archivist") to have permanent historical value in accordance with chapter 33 of title 44, United States Code.

(f) "Declassification guide" means written instructions issued by a declassification authority that describes the elements of information regarding a specific subject that may be declassified and the elements that must remain classified.

(g) "Downgrading" means a determination by a declassification authority that information classified and safeguarded at a specified level shall be classified and safeguarded at a lower level.

(h) "File series" means documentary material, regardless of its physical form or characteristics, that is arranged in accordance with a filing system or maintained as a unit because it pertains to the same function or activity.

Sec. 3.2. Authority for Declassification. (a) Information shall be declassified as soon as it no longer meets the standards for classification under this order.

(b) It is presumed that information that continues to meet the classification requirements under this order requires continued protection. In some exceptional cases, however, the need to protect such information may be outweighed by the public interest in disclosure of the information, and in these cases the information should be declassified. When such questions arise, they shall be referred to the agency head or the senior agency official. That official will determine, as an exercise of discretion, whether the public interest in disclosure outweighs the damage to national security that might reasonably be expected from disclosure. This provision does not:

(1) amplify or modify the substantive criteria or procedures for classification; or

(2) create any substantive or procedural rights subject to judicial review.

(c) If the Director of the Information Security Oversight Office determines that information is classified in violation of this order, the Director may require the information to be declassified by the agency that originated the classification. Any such decision by the Director may be appealed to the President through the Assistant to the President for National Security Affairs. The information shall remain classified pending a prompt decision on the appeal.

(d) The provisions of this section shall also apply to agencies that, under the terms of this order, do not have original classification authority, but had such authority under predecessor orders.

Sec. 3.3. Transferred Information. (a) In the case of classified information transferred in conjunction with a transfer of functions, and not merely for storage purposes, the receiving agency shall be deemed to be the originating agency for purposes of this order.

(b) In the case of classified information that is not officially transferred as described in paragraph (a), above, but that originated in an agency that has ceased to exist and for which there is no successor agency, each agency in possession of such information shall be deemed to be the originating agency for purposes of this order. Such information may be declassified or downgraded by the agency in possession after consultation with any other agency that has an interest in the subject matter of the information.

(c) Classified information accessioned into the National Archives and Records Administration ("National Archives") as of the effective date of this order shall be declassified or downgraded by the Archivist in accordance with this order, the directives issued pursuant to this order, agency declassification guides, and any existing procedural agreement between the Archivist and the relevant agency head.

(d) The originating agency shall take all reasonable steps to declassify classified information contained in records determined to have permanent historical value before they are accessioned into the National Archives. However, the Archivist may require that records containing classified information be accessioned into the National Archives when necessary to comply with the provisions of the Federal Records Act. This provision does not apply to information being transferred to the Archivist pursuant to section 2203 of title 44, United States Code, or information for which the National Archives and Records Administration serves as the custodian of the records of an agency or organization that goes out of existence.

(e) To the extent practicable, agencies shall adopt a system of records management that will facilitate the public release of documents at the time such documents are declassified pursuant to the provisions for automatic declassification in sections 1.6 and 3.4 of this order.

Sec. 3.4. Automatic Declassification. (a) Subject to paragraph (b), below, within 5 years from the date of this order, all classified information contained in records that (1) are more than 25 years old, and (2) have been determined to have permanent historical value under title 44, United States Code, shall be automatically declassified whether or not the records have been reviewed. Subsequently, all classified information in such records shall be automatically declassified no longer than 25 years from the date of its original classification, except as provided in paragraph (b), below.

(b) An agency head may exempt from automatic declassification under paragraph (a), above, specific information, the release of which should be expected to:

> **(1) reveal the identity of a confidential human source, or reveal information about the application of an intelligence source or method, or**

reveal the identity of a human intelligence source when the unauthorized disclosure of that source would clearly and demonstrably damage thenational security interests of the United States;

(2) reveal information that would assist in the development or use of weapons of mass destruction;

(3) reveal information that would impair U.S. cryptologic systems or activities;

(4) reveal information that would impair the application of state of the art technology within a U.S. weapon system;

(5) reveal actual U.S. military war plans that remain in effect;

(6) reveal information that would seriously and demonstrably impair relations between the United States and a foreign government, or seriously and demonstrably undermine ongoing diplomatic activities of the United States;

(7) reveal information that would clearly and demonstrably impair the current ability of United States Government officials protect the President, Vice President, and other officials for whom protection services, in the interest of national security, are authorized;

(8) reveal information that would seriously and demonstrably impair current national security emergency preparedness plans; or

(9) violate a statute, treaty, or international agreement.

(c) No later than the effective date of this order, an agency head shall notify the President through the Assistant to the President for National Security Affairs of any specific file series of records for which a review or assessment has determined that the information within those file series almost invariably falls within one or more of the exemption categories listed in paragraph (b), above, and which the agency proposes to exempt from automatic declassification. The notification shall include:

(1) a description of the file series;

(2) an explanation of why the information within the file series is almost invariably exempt from automatic declassification and why the information must remain classified for a longer period of time; and

(3) except for the identity of a confidential human source or a human intelligence source, as provided in paragraph (b), above, a specific date or event for declassification of the information.

The President may direct the agency head not to exempt the file series or to declassify the information within that series at an earlier date than recommended.

(d) At least 180 days before information is automatically declassified under this section, an agency head or senior agency official shall notify the Director of the Information Security Oversight Office, serving as Executive Secretary of the Interagency Security Classification Appeals Panel, of any specific information beyond that included in a notification to the President under paragraph (c), above, that the agency proposes to exempt from automatic declassification. The notification shall include:

(1) a description of the information;

(2) an explanation of why the information is exempt from automatic declassification and must remain classified for a longer period of time; and

(3) except for the identity of a confidential human source or a human intelligence source, as provided in paragraph (b), above, a specific date or event for declassification of theinformation. The Panel may direct the agency not to exempt the information or to declassify it at an earlier date than recommended. The agency head may appeal such a decision to the President through the Assistant to the President for National Security Affairs. The information will remain classified while such an appeal is pending.

(e) No later than the effective date of this order, the agency head or senior agency official shall provide the Director of the Information Security Oversight Office with a plan for compliance with the requirements of this section, including the establishment of interim target dates. Each such plan shall include the requirement that the agency declassify at least 15 percent of the records affected by this section no later than 1 year from the effective date of this order, and similar commitments for subsequent years until the effective date for automatic declassification.

(f) Information exempted from automatic declassification under thissection shall remain subject to the mandatory and systematicdeclassification review provisions of this order. (g) The Secretary of State shall determine when the United Statesshould commence negotiations with the appropriate officials of a foreigngovernment or international organization of governments to modify any treaty or international agreement that requires the classification of information

contained in records affected by this section for a period longer than 25 years from the date of its creation, unless the treaty or international agreement pertains to information that may otherwise remain classified beyond 25 years under this section.

Sec. 3.5. Systematic Declassification Review. (a) Each agency that has originated classified information under this order or its predecessors shall establish and conduct a program for systematic declassification review. This program shall apply to historically valuable records exempted from automatic declassification under section 3.4 of this order. Agencies shall prioritize the systematic review of records based upon:

(1) recommendations of the Information Security Policy Advisory Council, established in section 5.5 of this order, on specific subject areas for systematic review concentration; or

(2) the degree of researcher interest and the likelihood of declassification upon review.

(b) The Archivist of the shall conduct a systematic declassification review program for classified information: (1) acessioned into the National Archives as of the effective date of this order; (2) information transferred to the Archivist pursuant to section 2203 of title 44, United States Code; and (3) information for which the National Archives and Records Administration serves as the custodian of the records of an agency or organization that has gone out of existence. This program shall apply to pertinent records no later than 25 years from the date of their creation. The Archivist shall establish riorities for the systematic review of these records based upon the ecommendations of the Information Security Policy Advisory Council; or the degree of researcher interest and the likelihood of declassification upon review. These records shall be reviewed in accordance with the standards of this order, its implementing directives, and declassification guides provided to the Archivist by each agency that originated the records. The Director of the Information Security Oversight Office shall assure that agencies provide the Archivist with adequate and current declassification guides.

(c) After consultation with affected agencies, the Secretary of Defense may establish special procedures for systematic review for declassification of classified cryptologic information, and the Director of Central Intelligence may establish special procedures for systematic review for declassification of classified information pertaining to intelligence activities (including special activities), or intelligence sources or methods.

Sec. 3.6. Mandatory Declassification Review. (a) Except as provided in paragraph (b), below, all information classified under this order or predecessor orders shall be subject to a review for declassification by the originating agency if:

(1) the request for a review describes the document or material containing the information with sufficient specificity to enable the agency to locate it with a reasonable amount of effort;

(2) the information is not exempted from search and review under the Central Intelligence Agency Information Act; and

(3) the information has not been reviewed for declassification within the past 2 years. If the agency has reviewed the information within the past 2 years, or the information is the subject of pending litigation, the agency shall inform the requester of this fact and of the requester's appeal rights.

(b) Information originated by:

(1) the incumbent President;

(2) the incumbent President's White House Staff;

(3) committees, commissions, or boards appointed by the incumbent President; or

(4) other entities within the Executive Office of the President that solely advise and assist the incumbent President is exempted from the provisions of paragraph (a), above. However, the Archivist shall have the authority to review, downgrade, and declassify information of former Presidents under the control of the Archivist pursuant to sections 2107, 2111, 2111 note, or 2203 of title 44, United States Code. Review procedures developed by the Archivist shall provide for consultation with agencies having primary subject matter interest and shall be consistent with the provisions of applicable laws or lawful agreements that pertain to the respective Presidential papers or records. Agencies with primary subject matter interest shall be notified promptly of the Archivist's decision. Any final decision by the Archivist may be appealed by the requester or an agency to the Interagency Security Classification Appeals Panel. The information shall remain classified pending a prompt decision on the appeal.

c) Agencies conducting a mandatory review for declassification shall declassify information that no longer meets the standards for classification under this order. They shall release this information unless withholding is otherwise authorized and warranted under applicable law.

(d) In accordance with directives issued pursuant to this order, agency heads shall develop procedures to process requests for the mandatory review of classified information. These procedures shall apply to information classified under this or predecessor orders. They also shall provide a means for administratively appealing a denial of a mandatory review request, and for notifying the requester of the right to appeal a final agency decision to the Interagency Security Classification Appeals Panel.

(e) After consultation with affected agencies, the Secretary of Defense shall develop special procedures for the review of cryptologic information, the Director of Central Intelligence shall develop special procedures for the review of information pertaining to intelligence activities (including special activities), or intelligence sources or methods, and the Archivist shall develop special procedures for the review of information accessioned into the National Archives.

Sec. 3.7. Processing Requests and Reviews. In response to a request for information under the Freedom of Information Act, the Privacy Act of 1974, or the mandatory review provisions of this order, or pursuant to the automatic declassification or systematic review provisions of this order:

(a) An agency may refuse to confirm or deny the existence or nonexistence of requested information whenever the fact of its existence or nonexistence is itself classified under this order.

(b) When an agency receives any request for documents in its custody that contain information that was originally classified by another agency, or comes across such documents in the process of the automatic declassification or systematic review provisions of this order, it shall refer copies of any request and the pertinent documents to the originating agency for processing, and may, after consultation with the originating agency, inform any requester of the referral unless such association is itself classified under this order. In cases in which the originating agency determines in writing that a response under paragraph (a), above, is required, the referring agency shall respond to the requester in accordance with that paragraph.

Sec. 3.8. Declassification Database. (a) The Archivist in conjunction with the Director of the Information Security Oversight Office and those agencies that originate classified information, shall establish a Governmentwide database of information that has been declassified. The Archivist shall also explore other possible uses of technology to facilitate the declassification process.

(b) Agency heads shall fully cooperate with the Archivist in these efforts.

(c) Except as otherwise authorized and warranted by law, all declassified information contained within the database established under paragraph (a), above, shall be available to the public.

PART 4 SAFEGUARDING

Sec. 4.1. Definitions. For purposes of this order: (a) "Safeguarding" means measures and controls that are prescribed to protect classified information.

(b) "Access" means the ability or opportunity to gain knowledge of classified information.

(c) "Need-to-know" means a determination made by an authorized holder of classified information that a prospective recipient requires access to specific classified information in order to perform or assist in a lawful and authorized governmental function.

(d) "Automated information system" means an assembly of computer hardware, software, or firmware configured to collect, create, communicate, compute, disseminate, process, store, or control data or information.

(e) "Integrity" means the state that exists when information is unchanged from its source and has not been accidentally or intentionally modified, altered, or destroyed.

(f) "Network" means a system of two or more computers that can exchange data or information.

(g) "Telecommunications" means the preparation, transmission, or communication of information by electronic means.

(h) "Special access program" means a program established for a specific class of classified information that imposes safeguarding and access requirements that exceed those normally required for information at the same classification level.

Sec. 4.2. General Restrictions on Access. (a) A person may haveaccess to classified information provided that:

(1) a favorable determination of eligibility for access has been made by an agency head or the agency head's designee;

(2) the person has signed an approved nondisclosure agreement; and

(3) the person has a need-to-know the information.

(b) Classified information shall remain under the control of the originating agency or its successor in function. An agency shall not disclose information originally classified by another agency without its authorization. An official or employee leaving agency service may not remove classified information from the agency's control.

(c) Classified information may not be removed from official premises without proper authorization.

(d) Persons authorized to disseminate classified information outside the executive branch shall assure the protection of the information in a manner equivalent to that provided within the executive branch.

(e) Consistent with law, directives, and regulation, an agency head or senior agency official shall establish uniform procedures to ensure that automated information systems, including networks and telecommunications systems, that collect, create, communicate, compute, disseminate, process, or store classified information have controls that:

(1) prevent access by unauthorized persons; and

(2) ensure the integrity of the information.

(f) Consistent with law, directives, and regulation, each agency head or senior agency official shall establish controls to ensure that classified information is used, processed, stored, reproduced, transmitted, and destroyed under conditions that provide adequate protection and prevent access by unauthorized persons.

(g) Consistent with directives issued pursuant to this order, an agency shall safeguard foreign government information under standards that provide a degree of protection at least equivalent to that required by the government or international organization of governments that furnished the information. When adequate to achieve equivalency, these standards may be less restrictive than the safeguarding standards that ordinarily apply to United States "Confidential" information, including allowing access to individuals with a need-to-know who have not otherwise been cleared for access to classified information or executed an approved nondisclosure agreement.

(h) Except as provided by statute or directives issued pursuant to this order, classified information originating in one agency may not be disseminated outside any other agency to which it has been made available without the consent of the originating agency. An agency head or senior agency official may waive this requirement for specific information originated within that agency. For purposes of this section, the Department of Defense shall be considered one agency.

Sec. 4.3. Distribution Controls. (a) Each agency shall establish controls over the distribution of classified information to assure that it is distributed only to organizations or individuals eligible for access who also have a need-to-know the information.

(b) Each agency shall update, at least annually, the automatic,routine, or recurring distribution of classified information that they distribute. Recipients shall cooperate fully with distributors who are updating distribution lists and shall notify distributors whenever a relevant change in status occurs.

Sec. 4.4. Special Access Programs. (a) Establishment of special access programs. Unless otherwise authorized by the President, only the Secretaries of State, Defense and Energy, and the Director of Central Intelligence, or the principal deputy of each, may create a special access program. For special access programs pertaining to intelligence activities (including special activities, but not including military operational, strategic and tactical programs), or intelligence sources or methods, this function will be exercised by the Director of Central Intelligence. These officials shall keep the number of these programs at an absolute minimum, and shall establish them only upon a specific finding that:

(1) the vulnerability of, or threat to, specific information is exceptional; and

(2) the normal criteria for determining eligibility for access applicable to information classified at the same levelare not deemed sufficient to protect the information from unauthorized disclosure;

or

(3) the program is required by statute.

(b) Requirements and Limitations. (1) Special access programs shall be limited to programs in which the number of persons who willhave access ordinarily will be reasonably small and commensurate with the objective of providing enhanced protection for the information involved.

(2) Each agency head shall establish and maintain a system of accounting for special access programs consistent with directives issued pursuant to this order.

(3) Special access programs shall be subject to the oversight program established under section 5.6(c) of this order. In addition, the Director of the Information Security Oversight Office shall be afforded access to these programs, in accordance with the security requirements of each program, in order to perform the functions assigned to the Information Security Oversight Office under this order. An agency head may limit access to a special access program to the Director and no more than one other employee of the Information Security Oversight Office; or, for special access programs that are extraordinarily sensitive and vulnerable, to the Director only.

(4) The agency head or principal deputy shall review annually each special access program to determine whether it continues to meet the requirements of this order.

(5) Upon request, an agency shall brief the Assistant to the President for National Security Affairs, or his or her designee, on any or all of the agency's special access programs.

(c) Within 180 days after the effective date of this order, each agency head or principal deputy shall review all existing special access programs under the agency's jurisdiction. These officials shall terminate any special access programs that do not clearly meet the provisions of this order. Each existing special access program that an agency head or principal deputy validates shall be treated as if it were established on the effective date of this order.

(d) Nothing in this order shall supersede any requirement made by or under 10 U.S.C. 119.

Sec. 4.5. Access by Historical Researchers and Former Presidential Appointees. (a) The requirement in section 4.2(a)(3) of this order that access to classified information may be granted only to individuals who have a need-to-know the information may be waived for persons who:

(1) are engaged in historical research projects; or

(2) previously have occupied policy-making positions to which they were appointed by the President.

(b) Waivers under this section may be granted only if the agency head or senior agency official of the originating agency:

(1) determines in writing that access is consistent with the interest of national security;

(2) takes appropriate steps to protect classified information from unauthorized disclosure or compromise, and ensures that the information is safeguarded in a manner consistent with this order; and

(3) limits the access granted to former Presidential appointees to items that the person originated, reviewed, signed, or received while serving as a Presidential appointee.

PART 5 IMPLEMENTATION AND REVIEW

Sec. 5.1. Definitions. For purposes of this order: (a) "Self-inspection" means the internal review and evaluation of individual agency activities and the agency as a whole with respect to the implementation of the program established under this order and its implementing directives.

(b) "Violation" means:

(1) any knowing, willful, or negligent action that could reasonably be expected to result in an unauthorized disclosure of classified information;

(2) any knowing, willful, or negligent action to classify or continue the classification of information contrary to the requirements of this order or its implementing directives; or

(3) any knowing, willful, or negligent action to create or continue a special access program contrary to the requirements of this order.

(c) "Infraction" means any knowing, willful, or negligent action contrary to the requirements of this order or its implementing directives that does not comprise a "violation," as defined above.

Sec. 5.2. Program Direction. (a) The Director of the Office of Management and Budget, in consultation with the Assistant to the President for National Security Affairs and the co-chairs of the Security Policy Board, shall issue such directives as are necessary to implement this order. These directives shall be binding upon the agencies. Directives issued by the Director of the Office of Management and Budget shall establish standards for:

(1) classification and marking principles;

(2) agency security education and training programs;

(3) agency self-inspection programs; and

(4) classification and declassification guides.

(b) The Director of the Office of Management and Budget shall delegate the implementation and monitorship functions of this program to the Director of the Information Security Oversight Office.

(c) The Security Policy Board, established by a Presidential Decision Directive, shall make a recommendation to the President through the Assistant to the President for National Security Affairs with respect to the issuance of a Presidential directive on safeguarding classified information. The Presidential directive shall pertain to the handling, storage, distribution, transmittal, and destruction of and accounting for classified information.

Sec. 5.3. Information Security Oversight Office. (a) There is established within the Office of Management and Budget an Information Security Oversight Office. The Director of the Office of Management and Budget shall appoint the Director of the Information Security Oversight Office, subject to the approval of the President.

(b) Under the direction of the Director of the Office of Management and Budget acting in consultation with the Assistant to the President for National Security Affairs, the Director of the Information Security Oversight Office shall:

(1) develop directives for the implementation of this order;

(2) oversee agency actions to ensure compliance with this order and its implementing directives;

(3) review and approve agency implementing regulations and agency guides for

systematic declassification review prior to their issuance by the agency;

(4) have the authority to conduct on-site reviews of each agency's program established under this order, and to require of each agency those reports, information, and other cooperation that may be necessary to fulfill its responsibilities. If granting access to specific categories of classified information would pose an exceptional national security risk, the affected agency head or the senior agency official shall submit a written justification recommending the denial of access to the Director of the Office of Management and Budget within 60 days of the request for access. Access shall be denied pending a prompt decision by the Director of the Office of Management and Budget, who shall consult on this decision with the Assistant to the President for National Security Affairs;

(5) review requests for original classification authority from agencies or officials not granted original classification authority and, if deemed appropriate, recommend Presidential approval through the Director of the Office of Management and Budget;

(6) consider and take action on complaints and suggestions from persons within or outside the Government with respect to the administration of the program established under this order;

(7) have the authority to prescribe, after consultation with affected agencies, standardization of forms or procedures that will promote the implementation of the program established under this order;

(8) report at least annually to the President on the implementation of this order; and

(9) convene and chair interagency meetings to discuss matters pertaining to the program established by this order.

Sec. 5.4. Interagency Security Classification Appeals Panel. (a) Establishment and Administration.

(1) There is established an Interagency Security Classification Appeals Panel ("Panel"). The Secretaries of State and Defense, the Attorney General, the Director of Central Intelligence, the Archivist of the United States, and the Assistant to the President for National Security Affairs shall each appoint a senior level representative to serve as a member of the Panel. The President shall select the Chair of the Panel from among the Panel members.

(2) A vacancy on the Panel shall be filled as quickly as possible as provided in paragraph (1), above.

(3) The Director of the Information Security Oversight Office shall serve as the Executive Secretary. The staff of the Information Security Oversight Office shall provide program and administrative support for the Panel.

(4) The members and staff of the Panel shall be required to meet eligibility for access standards in order to fulfill the Panel's functions.

(5) The Panel shall meet at the call of the Chair. The Chair shall schedule meetings as may be necessary for the Panel to fulfill its functions in a timely manner.

(6) The Information Security Oversight Office shall include in its reports to the President a summary of the Panel's activities.

(b) Functions. The Panel shall:

(1) decide on appeals by persons who have filed classification challenges under section 1.9 of this order;

(2) approve, deny, or amend agency exemptions from automatic declassification as provided in section 3.4 of this order; and

(3) decide on appeals by persons or entities who have filed requests for mandatory declassification review under section 3.6 of this order.

(c) Rules and Procedures. The Panel shall issue bylaws, which shall be published in the Federal Register no later than 120 days from the effective date of this order. The bylaws shall establish the rules and procedures that the Panel will follow in accepting, considering, and issuing decisions on appeals. The rules and procedures of the Panel shall provide that the Panel will consider appeals only on actions in which: (1) the appellant has exhausted his or her administrative remedies within the responsible agency; (2) there is no current action pending on the issue within the federal courts; and (3) the information has not been the subject of review by the federal courts or the Panel within the past 2 years.

(d) Agency heads will cooperate fully with the Panel so that it can fulfill its functions in a timely and fully informed manner. An agency head may appeal a decision of the Panel to the President through the Assistant to the President for National Security Affairs. The Panel will report to the President through the Assistant to the President for National Security Affairs any instance in which it believes that an agency head is not cooperating fully with the Panel.

(e) The Appeals Panel is established for the sole purpose of advising and assisting the President in the discharge of his constitutional and discretionary authority to protect the national security of the United States. Panel decisions are committed to the discretion of the Panel, unless reversed by the President.

Sec. 5.5. Information Security Policy Advisory Council. (a) Establishment. There is established an Information Security Policy Advisory Council ("Council"). The Council shall be composed of seven members appointed by the President for staggered terms not to exceed 4 years, from among persons who have demonstrated interest and expertise in an area related to the subject matter of this order and are not otherwise employees of the Federal Government. The President shall appoint the Council Chair from among the members. The Council shall comply with the Federal Advisory Committee Act, as amended, 5 U.S.C. App. 2.

(b) Functions. The Council shall:

(1) advise the President, the Assistant to the President for National Security Affairs, the Director of the Office of Management and Budget, or such other executive branch officials as it deems appropriate, on policiesestablished under this order or its implementing directives, including recommended changes to those policies;

(2) provide recommendations to agency heads for specific subject areas for systematic declassification review; and

(3) serve as a forum to discuss policy issues in dispute.

(c) Meetings. The Council shall meet at least twice each calendar year, and as determined by the Assistant to the President for National security Affairs or the Director of the Office of Management and Budget.

(d) Administration.

(1) Each Council member may be compensated at a rate of pay not to exceed the daily equivalent of the annual rate of basic pay in effect for grade GS-18 of the general

schedule under section 5376 of title 5, United States Code, for each day during which that member is engaged in the actual performance of the duties of the Council.

(2) While away from their homes or regular place of business in the actual performance of the duties of the Council, members may be allowed travel expenses, including per diem in lieu of subsistence, as authorized by law for persons serving intermittently in the Government service (5 U.S.C. 5703(b)).

(3) To the extent permitted by law and subject to the availability of funds, the Information Security Oversight Office shall provide the Council with administrative services, facilities, staff, and other support services necessary for the performance of its functions.

(4) Notwithstanding any other Executive order, the functions of the President under the Federal Advisory Committee Act, as amended, that are applicable to the Council, except that of reporting to the Congress, shall be performed by the Director of the Information Security Oversight Office in accordance with the guidelines and procedures established by the General Services Administration.

Sec. 5.6. General Responsibilities. Heads of agencies that originate or handle classified information shall: (a) demonstrate personal commitment and commit senior management to the successful implementation of the program established under this order;

(b) commit necessary resources to the effective implementation of the program established under this order; and

(c) designate a senior agency official to direct and administer the program, whose responsibilities shall include:

(1) overseeing the agency's program established under this order, provided, an agency head may designate a separate official to oversee special access programs authorized under this order. This official shall provide a full accounting of the agency's special access programs at least annually;

(2) promulgating implementing regulations, which shall be published in the Federal Register to the extent that they affect members of the public;

(3) establishing and maintaining security education and training programs;

(4) establishing and maintaining an ongoing self-inspection program, which shall include the periodic review and assessment of the agency's classified product;

(5) establishing procedures to prevent unnecessary access to classified information, including procedures that: (i) require that a need for access to classified information is established before initiating administrative clearance procedures; and (ii) ensure that the number of persons granted access to classified information is limited to the minimum consistent with operational and security requirements and needs;

(6) developing special contingency plans for the safeguarding of classified information used in or near hostile or potentially hostile areas;

(7) assuring that the performance contract or other system used to rate civilian or military personnel performance includes the management of classified information as a critical element or item to be evaluated in the rating of: (i) original classification authorities; (ii) security managers or security specialists; and (iii) all other personnel whose duties significantly involve the creation or handling of classified information;

(8) accounting for the costs associated with the implementation of this order, which shall be reported to the Director of the Information Security Oversight Office for publication; and

(9) assigning in a prompt manner agency personnel to respond to any request, appeal, challenge, complaint, or suggestion arising out of this order that pertains to classified information that originated in a component of the agency that no longer exists and for which there is no clear successor in function.

Sec. 5.7. Sanctions. (a) If the Director of the Information Security Oversight Office finds that a violation of this order or its implementing directives may have occurred, the Director shall make a report to the head of the agency or to the senior agency official so that corrective steps, if appropriate, may be taken.

(b) Officers and employees of the United States Government, and its contractors, licensees, certificate holders, and grantees shall be subject to appropriate sanctions if they knowingly, willfully, or negligently:

(1) disclose to unauthorized persons information properly classified under this order or predecessor orders;

(2) classify or continue the classification of information in violation of this order or any implementing directive;

(3) create or continue a special access program contrary to the requirements of this order; or

(4) contravene any other provision of this order or its implementing directives.

(c) Sanctions may include reprimand, suspension without pay, removal, termination of classification authority, loss or denial of access to classified information, or other sanctions in accordance with applicable law and agency regulation.

(d) The agency head, senior agency official, or other supervisory official shall, at a minimum, promptly remove the classification authority of any individual who demonstrates reckless disregard or a pattern of error in applying the classification standards of this order.

(e) The agency head or senior agency official shall:

(1) take appropriate and prompt corrective action when a violation or infraction under paragraph (b), above, occurs; and

(2) notify the Director of the Information Security Oversight Office when a violation under paragraph (b)(1), (2) or (3), above, occurs.

PART 6 GENERAL PROVISIONS

Sec. 6.1. General Provisions. (a) Nothing in this order shall supersede any requirement made by or under the Atomic Energy Act of 1954, as amended, or the National Security Act of 1947, as amended. "Restricted Data" and "Formerly Restricted Data" shall be handled, protected, classified, downgraded, and declassified in conformity with the provisions of the Atomic Energy Act of 1954, as amended, and regulations issued under that Act.

(b) The Attorney General, upon request by the head of an agency or the Director of the Information Security Oversight Office, shall render an interpretation of this order with respect to any question arising in the course of its administration.

(c) Nothing in this order limits the protection afforded any information by other provisions of law, including the exemptions to the Freedom of Information Act, the Privacy Act, and the National Security Act of 1947, as amended. This order is not intended, and should not be construed, to create any right or benefit, substantive or procedural, enforceable at law by a party against the United States, its agencies, its officers, or its employees. The foregoing is in addition to the specific provisos set forth in sections 1.2(b), 3.2(b) and 5.4(e) of this order.

(d) Executive Order No. 12356 of April 6, 1982, is revoked as of the effective date of this order.

Sec. 6.2. Effective Date. This order shall become effective 180 days from the date of this order.

WILLIAM J. CLINTON

THE WHITE HOUSE, April 17, 1995

MAKING THE TRANSITION FROM DATA MANAGEMENT TO INFORMATION ASSET MANAGEMENT - LESSONS LEARNED WITHIN THE LAW ENFORCEMENT, INTELLIGENCE AND DEFENSE COMMUNITIES

by Barbara J. Dutton
James Martin Government Intelligence, Inc.

1. INTRODUCTION

Agencies within the law enforcement, intelligence and defense communities have traditionally viewed themselves as having a service mission.

> *"To uphold the law in a manner that is faithful to the Constitution of the United States."* The FBI Mission

> *"The Defense Logistics Agency is a combat support agency responsible for worldwide logistics support throughout the Department of Defense. The primary focus of the Agency is to support the warfighter in time of war and in peace, and to provide relief efforts during times of national emergency."* The Defense Logistics Agency Mission

However, they can be viewed as very much being in the information business. The FBI does not actually convict a felon of acting against the Constitution or the national security of the United States. Rather, it collects information about suspected crimes and criminals, analyzes that information to prove or disprove a crime, packages the information in various ways, and delivers it externally to its "customers" for use in prosecution or national security threat neutralization. Seen in this perspective, agency operational activities are focused on acquiring, analyzing, managing and packaging information to supply information products and services, both for internal consumption and for external customers.

These agencies have long recognized capital, people and equipment as critical resources to fulfilling their mission. Within these communities, it is relatively recent that information has become to be viewed as a critical agency resource, or asset, and primary product. For this to happen, the direct contribution of information as a strategic asset to the mission, goals/objectives and institutional knowledge of an agency must be well understood. Additionally, the Information Management organization's responsibility of building automated systems becomes viewed as building critical information tools required to support the core mission of the organization. The FBI's Information Management organization could be viewed as building investigative information tools to support the primary mission of the organization -- investigations. Such a change in perspective, deceptively simple, is actually a radical departure from viewing information as a strategic agency asset versus data.

To manage and package important information assets and to build essential core information tools requires a new framework different than traditional data management. This framework is Information Asset Management.

Information assets are any type of "object" describing or having information value. They range from objects describing customers and their satisfiers, products and services, to software and technical infrastructure components. Information assets refer to very simple objects, such as the modeling structures of entity types, relationships, object classes and messages, to complex objects, such as technology with embedded software components.

This paper describes experiences of consulting with several agencies within federal law enforcement, intelligence and defense communities as they made the decision to and subsequently transitioned from a traditional data management to Information Asset Management paradigm.

2. THE MANDATE FOR CHANGE

Agencies within law enforcement, intelligence and defense are facing an increasing amount and diversity of change. National Performance Review legislation, the elimination of communist political and Soviet military threats, and increasing public concern with and the globalization of crime and national security threats are causing these communities to redefine their very existence. At a minimum, they are under pressure to better respond to public concern and increase "customer" satisfaction.

What Does the Future Look Like to These Communities?

Characteristics of the future reality for these agencies include:

- fewer resources
- increasing competition for resources (internally within an agency; and externally competing with other agencies for Congressional budget)
- requirement to do more with less
- changing crime trends
- changing criminal organizations
- globalization of crime/national security threats
- fluidity of virtual agencies (new partnerships)
- explosion of technology
- faster rate of change
- increasing societal diversity
- shift from military target warfare and the ground combative soldier, towards information warfare and the information warrior

What is Their History?

Some aspects of these long-standing agencies are dramatically changing, while other aspects remain constant.

- The direction and leadership style exhibited and required at many agencies has changed from being despotic to one of providing a motivating strategic vision and supportive agency leadership and rejuvenation while establishing stronger inter-agency partnerships.

- Resources have changed from being predominantly white male to increasingly diverse in culture and ethnicity.

- Most agency values have remained fairly constant. Fidelity, bravery and integrity have always been FBI values.

- Management has been based on hierarchical structures and chain of command practices. This is still the predominant model, however, Total Quality Management initiatives have introduced new management models to federal government.

Many federal agencies are literally fighting for their very survival. Such agencies include the US Agency for International Development, Bureau of Alcohol, Tobacco and Firearms, Drug Enforcement Agency, US Postal Service, and all military bases. Reasons for their struggle are varied. In the case of the US Postal Service, it is both performance and competition based. Public pressure and changing national priorities have caused the reevaluation of US Agency for International Development programs. Realignment and base closures are in response to changes in fundamental agency goals and objectives and decreasing available resources. In all cases, there has been some aspect of customer satisfaction, or lack thereof, that has created a crisis for federal government today.

The government can't give job security, only customers can.

It has been suggested that every surviving agency will have an inordinate dependence on their information systems. However, their Information Management organizations will have to undergo extraordinary change before they can be expected to make a noticeable contribution. Information Management organizations have long promised to deliver system solutions to support rapid implementation of new agency ventures. These organizations have shown disappointing results in most cases. The technological challenges confronting these organizations are not the cause of their inability to achieve industry superiority. Rather, the primary reasons have been more cultural and political.

3. CHARACTERISTICS OF THE INFORMATION ASSET MANAGEMENT FRAMEWORK

Information is a resource critical to conducting every activity within an organization. Information Assets are common and critical to all agency components. This is true from several perspectives. One, information about strategic agency assets, their requirements, utilization and inter-dependencies, is critical to achieve objectives and to responsibly utilize resources in today's environment of scarce resources.

Two, information is required to conduct any agency activity. That information, or better yet, knowledge, must be available in a timely manner appropriate to the activities being carried out and the decisions being made.

Three, all activities within an organization should ultimately contribute to providing value to a customer, to fulfilling the mission of the organization and to meeting goals and objectives. These activities, or processes, can be identified and described in process profiles. These process profiles provide a model of the business, especially as it relates to customers, either internal or external. These models must be managed and developed in an integrated fashion. It is the information about the processes, not the processes themselves, which constitute information assets, and which require an integrated approach to manage the way they are conceptualized, defined , the way they are improved, and the information tools which are developed to support them. Information Asset Management, as described in this document, focuses on managing and providing information assets as the means for enabling the integration and coordination of organizational activities.

The primary concepts underlying the Information Asset Management framework implemented by client agencies are presented below. They are contrasted to traditional data management concepts.

Information Asset Management Concepts

Objects. The concept of "objects" is critical to the Information Asset Management Framework. An object is equivalent to an information asset, and is loosely defined to be anything about which information is collected, from both a data or process perspective. Objects are integrated at various levels of aggregation to build information tools in direct support of an agency's business. They form the information products that an agency builds as a result of conducting that business. Objects are the actual information tools themselves, such as hardware, system software, and applications. They are the models behind such objects and the plans for building these objects.

It is valuable to refer to different information tool components as objects because the processes used to build and manage the objects are the same. Differences are introduced by some detailed procedures and detailed standards associated with object types.

Object profiles were developed to capture information used to support management decisions, to effect integration and reusability, and to assess change impacts. An example of an object profile is presented below.

Object Profile

Custodian Type
The IAM role responsible for maintaining the current definition of the object, or responsible for the security of the object

Technique
The methodology technique used to develop and document the object

Activity
The methodology activity employing a particular technique to produce an object deliverable

Aggregate of:
Indication of all lower-level objects comprising the higher level object

Component of:
Architecture - The architecture (Information/Systems/Technical) the object belongs to
Model - The model the object is contained in
Diagram - The diagram the object is documented in

Tool
The tool used to capture the documentation of the object

Policy
Any policy relevant to the object

Standards
Standards relating to the object

Quality Measure
The criteria for evaluating the quality of the object

Security Level
Indication of security requirements surrounding the object, or access to the object

Transforms to:
The other objects this object is transformed to

Transforms from:
The objects this object was transformed from

Synonym
The various names used to reference the same object

Change Process
The procedures to be followed to change the definition of an existing object

Information Location
A reference to documentation about this object, or to methodologies used to develop the object.

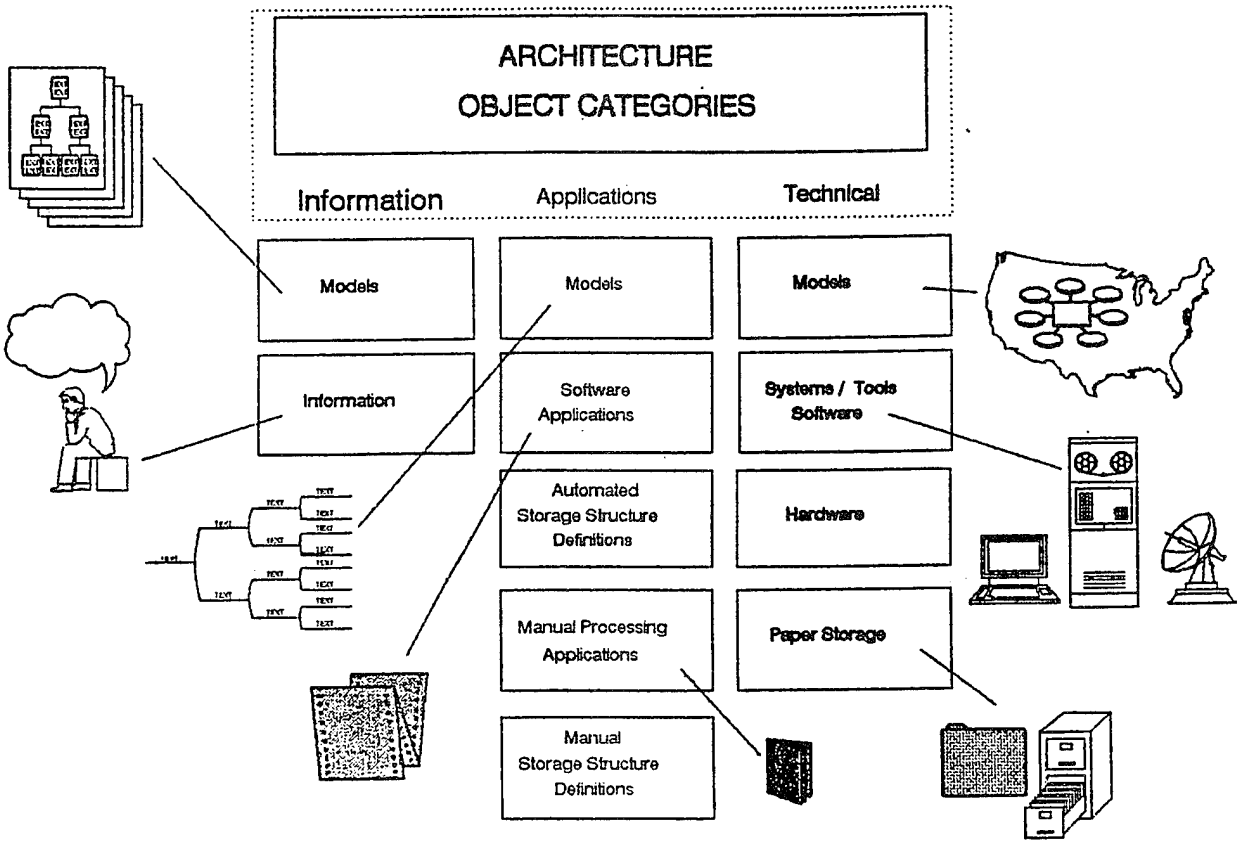
This view of information and way of managing information is in contrast to defining and managing simpler data objects within a traditional data management paradigm.

Architectures. Architectures are the highest level of objects representing information assets. Client agencies implemented three architectures--information, systems and technical. The Information Architecture represents a

logical picture of the business. The Systems Architecture represents a physical picture of the business, in terms if its business systems. The Technical Architecture represents the technical infrastructure components supporting the business systems and enabling the business. The architectures are used to manage the models and plans for architecture implementation, and the actual information assets or architecture objects themselves.

The Architecture Object Categories Diagram represents the highest level of object categories within the three architectures. The architectures include the plans for building information products and the actual information products themselves. They include current and future inventories. They include automated and manual objects.

The architectures provide for inter- and intra-architecture integration and interoperability vs. database integration and interoperability. This broadens the scope of management, from stakeholder through technology, versus just data.



Integrated Information Products. The Information Asset Management framework accommodates building integrated information products, encompassing data, process, technology and delivery mechanisms to an internal or external customer. The integration of object components required to produce information products can be identified across the three architectures.

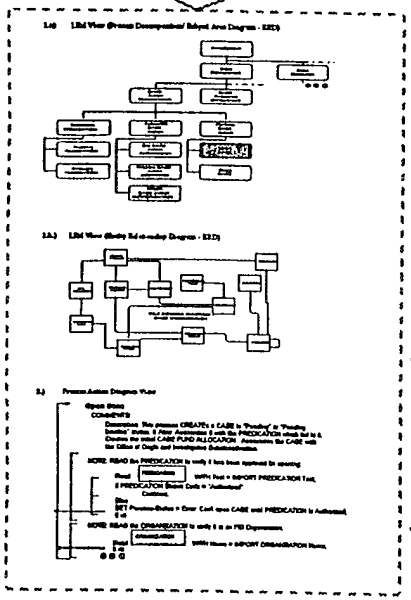
This is in contrast to a simpler view of data and database products.

Object Mapping. Object mapping is required to maintain integration within and across the architectures. As an object is changed, it is necessary to identify and distinguish between its versions. For example, different versions of a logical object may be needed to trace historical changes in enterprise functions or different versions of a physical object, such as a systems software package, may be required for different hardware platforms. Mapping tracks a logical object, in all its logical versions, to all versions of its physical manifestations and to other architecture objects supporting or depending on it. Mapping also supports establishing relationships between differing object structures, i.e., Information Engineering object structures and how they relate to Object Oriented object structures.

Examples of architecture integration defined through object mapping are shown in the following diagrams. This is in contrast to simpler mapping of data objects to each other, or of just logical to physical data object mapping.

Information Architecture

Activity, Data, & Interaction
- Decompostion Diagram
- Entity Relationship Diagram
- Process Action Diagrams
- Process Logic Diagrams
- Dependency Diagrams
- Strategic Plan



Architecture Interaction Model

Interaction Between the Information Architecture & Technical Architecture Model Objects ~
Activity, Data, & Interaction Analysis

Activities by Functional Area (Current)

Priorities, Mission, and Objectives

Activities by Functional Area (Future)

FAA Schedules & Preliminary Technology Requirements

Technology Initiatives Impacting the LIM

Technical Architecture

1.) Application Summary/Workload Characteristics

	ADABAS/ NIECSC	ADABAS/ NIECSC WERCSC	ADABAS/ NIECSC WERCSC

2.) Building Blocks

Functional Building Blocks – Trends in "Class 4" Workstations

Processor				
# Processors				
Memory				
Display				
Storage				
Capabilities				

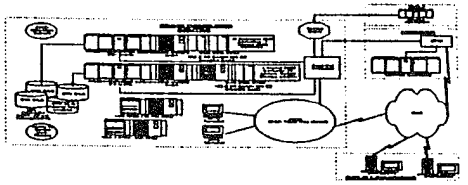
3.) Costing Model

Hardware Related Annual Costs	Year 1	Year 2	Year 3	Year 4	Year 5	Life Cycle
Mainframe Purchase						
Maintenance						
Mini Purchase						
Tape/Secondary Storage Purchase						
Maintenance						
ETC.						
Hardware Subtotals						

4.) Evaluation Criteria

EVALUATION FACTOR	RELATIVE IMPORTANCE OF FACTORS (Rate 1 to 6)	ELEMENTS WITHIN FACTOR	IMPORTANCE OF ELEMENTS (within Factor)
Interoperability			Rate 1 to 4
		Degree of application of OSI Protocols and Standards	
		Ease of interchange with heterogeneous environments	
		Ops for assembled homogeneous products	
		Supports Security requirements without compromising environment	
Ease of Implementation etc.			Rate 1 to 5
		Transparency to Users	

5.) Technical Architecture Schematics (Current, 2-, 5- and 10-year Views)



6.) Implementation Plan

Architecture Interaction Model

Interaction Between the Applications Architecture & Technical Architecture Model Objects ~ *Distribution/Frequency Information*

Applications Architecture

Distribution/Frequency Information
- Procedure Distribution Matrix
- Data Element Volume Matrix
- Application Workflows
- Application Descriptions

Workload Characteristics for Current/Future Applications

Application Priorities, Mission, and Objectives

Frequency Specifications Input into Future Application Processing & Storage

Application Distribution Evaluation

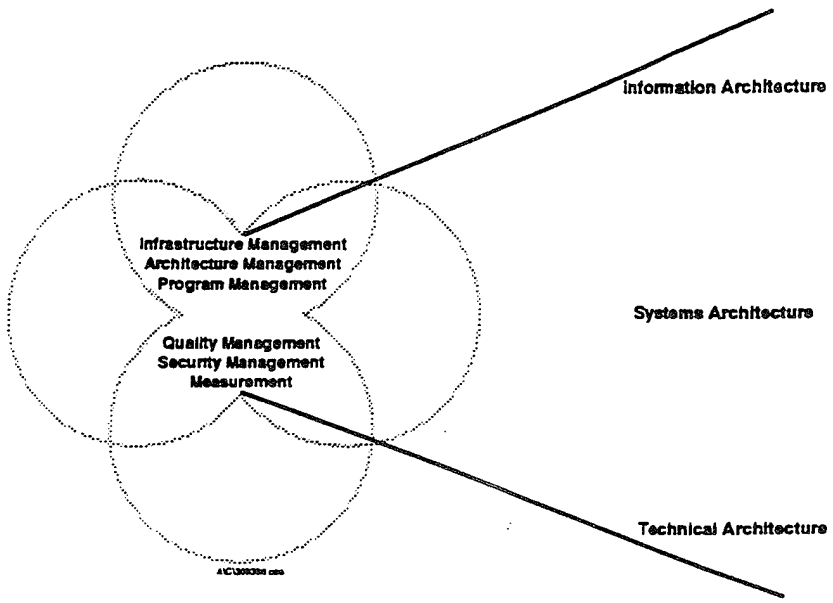
Distribution Specifications Impacting the Technical Architecture for Supporting Future Applications

BSD/BSI Schedules & Refined Technology Specifications

Technical Architecture

Information Asset Management Functions

Information Asset Management functions (a group of activities that together support one aspect of furthering the mission of the agency. A function describes what is done within the enterprise and is independent of the organization structure.) should collectively further the mission of the enterprise. The six high-level Information Asset Management functions by managing and providing access to the agency's information assets are shown in the Information Asset Management Functions diagram below.



Information Asset Management Functions Diagram

Infrastructure Management establishes an efficient and effective infrastructure required to support all IAM activities. This infrastructure includes the organizational design, training, methodologies and tools, and a transition plan to implement the IAM framework, articulating a cultural change management approach.

Architecture Management establishes the Information Infrastructure as the foundation for agency activities, and provides architectures as a common decision making framework and management tool to be used by all people involved in IAM activities. Reusability and integration are achieved through Architecture Management.

Program Management establishes strategic IAM objectives and maintains their alignment with the agency vision, mission and objectives, as well as evaluates overall program efficiency and effectiveness. Program Management acquires resources, and includes budget formulation, resource allocation, budget allocation, and tracks resources, including budget execution. These activities must ensure programmatic integration of IAM efforts.

Quality Management provides quality leadership; quality assurance of products, systems and services; and ensures that IAM processes are continually optimized.

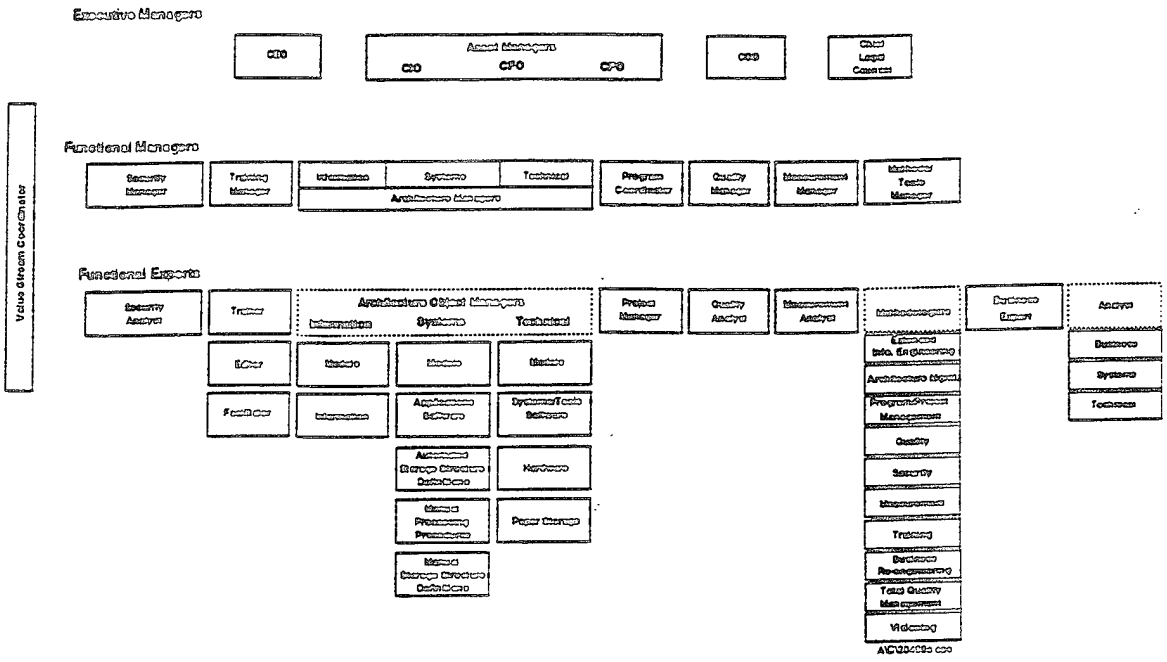
Security Management ensures information confidentiality, availability, and integrity, while ensuring information is distributed to the right people at the right time.

Measurement provides the quantitative and objectives basis for Quality Management, and for evaluating all aspects of IAM.

Information Asset Management Roles

Several roles must be established to carry out Information Asset Management functions. Each role is defined by a unique set of responsibilities, authority and expertise requirements. Many roles are already established roles within the enterprise which provide executive-level oversight.

The Information Asset Management Roles diagram presents core roles and their logical relationships to each other. This hierarchy does not imply an actual reporting structure nor the number of individuals fulfilling each role. Rather, it provides a logical design whose organizational implementation must be customized.



Information Asset Management Roles Diagram

Concept of Operations. At the first level of authority, the Executive Manger role of Chief Information Officer has the authority to make virtually all decisions regarding program objectives and program resources. The Architecture Sponsors represent the highest authority for business and technical requirements and priorities. Relatively few decisions in managing the program should need to be raised above this level for adjudication.

At the second level of authority, the Functional Managers provide a centralized focus and accountability for the work performed by Functional Experts who are distributed across the organization. The Functional Managers should work as a team to formulate strategic and tactical plans and resolve disputes within the architecture framework. The Functional Managers give direction to and empower the Functional Experts.

At the third level of authority, the Functional Experts are empowered with the knowledge, skills, and abilities to accomplish their work according to established policies, procedures, and standards. The Functional Experts can be distributed across the organization, and participate in multi-divisional project teams to accomplish defined project objectives. Within the project teams, certain Information Asset Management functions will be carried out by project team members as an integral part of their project work.

The Value Stream Manager spans levels to focus the work of and apply expertise to optimizing and support integrated agency operational processes and to produce integrated information products.

The Re-engineering of Data Management and Information Management Organizations

Implementing the Information Asset Management framework represents a re-engineering of the traditional data management function and of traditional Information Management organizations. This re-engineering is based on:

- a different perspective of data as a strategic asset of the enterprise, and of Information Asset Management as a strategic agency function directly contributing to the core mission of the agency;
- continuous change mechanisms as an inherent feature of information management practices to ensure optimum performance and to revalidate strategic and tactical focus;
- an understanding of the Information Management organization's customers and products/services which "delight" them.

4. LESSONS LEARNED

Consulting experiences with client agencies have indicated several lessons learned in transitioning from a data management to Information Asset Management framework. These lessons illuminate both benefits and issues to be dealt with.

Actual benefits of implemented Information Asset Management frameworks include enabling:

- a total quality perspective focused on customer satisfaction and building quality information products;
- reusability of architecture objects, including business models, software components, database design, and technical infrastructure components;
- package/object library acquisition;
- business process, data, system and technology integration;
- change management vs. configuration management;
- improvement of data management processes;
- rejuvenating of and stronger partnerships with the Information Management organization.

Transition issues include:

- lack of a shared vision;
- persistency of the old culture and complacency;
- inappropriate performance measures;
- failure to learn as an organization;
- lack of a framework relating information to agency mission;
- not recognizing the extent of change required to transition from a traditional data management paradigm to an Information Asset paradigm;
- acquiring new skills for existing staff;
- distributed decision making and consensus building;
- effecting an information vs. data or technology culture;
- developing business/technical partnerships;
- introducing new technologies, such as repositories, I-CASE, object libraries, work group tools;
- implementing a new organization design based on information asset vs. data roles;
- achieving two objectives simultaneously -- organizational transformation and product delivery;
- introduction of discipline and structure -- introduction of a methodology.

While the benefits from new tools and methodologies addressing Information Asset Management are significant, managing the transition was challenging. The transition was met with confusion and uncertainty, leading to resistance and the risk that the transition would fail. To effectively manage the development and usage of information assets, provisions were made to establish the necessary support infrastructure, including an effective organization design and management practices, methodology and tools expertise, training, and infrastructure transition planning which addressed cultural and social change issues. Once these were in place, Information Asset Management started to become a very natural part of the business.

Critical Success Factors for the Future

Several critical issues must be addressed in the near term to ensure organizations are positioned to meet the challenges of the future. Organizations must develop:

- a clear vision for the future, and strategic direction enabling realization of the vision;
- an understanding of its customers and products/services which "delight" them;
- strategies for utilizing information technology to leverage strategic advantage and competitiveness;
- a commitment to invest in critical strategic resources for the future, especially information assets;
- continuous change mechanisms as an inherent feature of business practices to ensure optimum performance and to revalidate strategic and tactical focus;
- an ability to respond quickly to change;
- instituting a culture which embraces change and is able to manage change effectively.

Factors critical to the success of Information Asset Management include the following.

Openness to change coupled with effective change management and change integration strategies.

Dynamic infrastructure flexible for change.

Philosophical approach to developing total business solutions, not just utilizing the latest techniques and tools.

Business and Technical Management Commitment/Involvement. Information Asset Management is a business function; its effectiveness directly depends on its perceived value by Executive Management. Therefore, it is critical that management be informed and committed so it understands and supports the purpose and objectives of Information Asset Management.

Management, especially the agency director and CIO, must keep the long-term benefits of Information Asset Management in view, and not just focus on achieving short-term benefits. It must set priorities for agency-wide objectives versus organization unit objectives. Management must set the tone for cooperation across organization units, and foster a spirit of working together for the good of the enterprise.

Management commitment is crucial to re-engineering the Information Management organization. Top management must set the stage for aggressive Business Re-engineering by creating an appetite for change. An evolving business climate often motivates management to pursue radical change. They generally accomplish this either by highlighting problems or stating a challenging vision. In either case, they must set a tone of urgency. This is especially crucial in successful organizations that may be faced with cultural rigidity and are wedded to today's approaches to success. If top management is lukewarm about the need for change, then the lower levels of management and employees will not engage in the radical reassessments that are the heart of successful Business Re-engineering.

Empowerment. In carrying out a management function, Information Asset Management roles must have the authority needed to meet their responsibilities, or subsequent benefits will not be achieved. Information Asset Management is not merely a monitoring and reporting function. Cross-functional teams must be empowered to make decisions appropriate to their responsibilities. Bureaucratic management levels not adding value must not impede progress.

Proper Tools. The Information Asset Management function is complex and requires extensive use of I-CASE technology and other supporting tools. Access to a repository and facilitating automated tools, including object management, program/project/process management, and metrics tools is required to support many Information Asset Management functions.

Training. A comprehensive training program is required to support the introduction of IAM and associated methods and tools. While training is not sufficient in and of itself, it helps ensure that individuals acquire the required skills, and are kept informed about new concepts and practices.

Information Dissemination. The advent of a new framework and associated methods and tools is surrounded by much information which must be assimilated by managers and practitioners alike. This information must be disseminated in a timely fashion, and made easily accessible. Coordinating IAM efforts requires effective communication between all roles and organization units involved. This communication must be facilitated by effective dissemination of information.

To meet these challenges requires taking an integrated approach to change.

5. SUMMARY

Preparing for the future is an urgent and difficult executive responsibility in today's turbulent government environment. Increasing amounts of management time and ingenuity are directed toward making existing organizations more maneuverable as a means of achieving competitive advantage. Goals such as customer focus, public driven quality and globalization often remain unfulfilled despite the most thoughtful, detailed and brilliantly communicated agency strategy. Many leaders attempting to prepare their agencies for the future discover to their frustration that substantial expenditures on planning and reorganization generate only trivial differences in performance.

It is no longer sufficient to take an incremental improvement approach to change--it is a requirement for survival. Radical change is required to move Information Management organizations out of their current old paradigms into a new framework for doing business. This is required to conduct the business of information management, as well as to be a true partner with the rest of an agency. The key to understanding Information Asset Management is to recognize is as a strategic agency function that must work in partnership with operational functions to offer total agency solutions. The CIO responsible for Information Asset Management is responsible for increasing the value of the enterprise's information assets, and for creating the Information Management organization for the future.

Organizations that have invested in information assets and are managing them from an agency perspective are poised to become the new leaders within their communities. Information Management organizations continuing to manage information within a traditional data management paradigm are facing declining profits and return on investment in information assets, ineffectiveness and extinction. There is a requirement for a new vision for information assets, and for management of them within a new framework.

The requirement for Information Asset Management is information; the opportunity is knowledge. The greatest power we have is the ability to envision our own fate--and to change ourselves.

6. BIOGRAPHICAL SKETCH

Ms. Dutton is currently Vice President of James Martin Government Intelligence, a premier international consulting firm offering products and services in strategic visioning, business re-engineering, total quality management and information technology. She has specialized in developing and managing information services and business improvement projects/programs during her professional career. She has worked in both international public and private sector organizations during that time. She is an innovator at creating new methodologies in both information distribution and information management, and has translated these to products and consulting services. She has helped organize and start two new consulting firms. Her management experience includes managing large Enterprise Engineering (business improvement and information services) programs in the public sector. Her consulting includes policy development, organization design and cultural change management, methodology and product development, and full life-cycle system development in a variety of technical environments. Her work has involved use of many different technologies, including I-CASE and repository tools.

Ms. Barbara J. Dutton, Vice President
James Martin Government Intelligence, Inc.
4350 North Fairfax Drive, Suite 610
Arlington, Virginia 22203
703-528-5515 phone
703-528-5546 fax
BJDutton@aol.com

Automating Information Exchange Between Self-Describing Databases

James Coleman and Lisa Sills
Georgia Institute of Technology
Georgia Tech Research Institute
Information Technology and Telecommunications Laboratory
Atlanta, Georgia 30332-0800

Abstract - Georgia Tech and the Army Research Laboratory, in a federated program, have proven the utility of applying self-describing database technology to the problem of information exchange between heterogeneous legacy and migration C4I systems. The initial capability provides for the automatic exchange of message data and new message formats directly from database to database employing a meta-grammar that captures the rules of the U.S. Message Text Formatting program (USMTF). This allows new formats to migrate automatically to where they are needed and provides the means of direct access of that data by the client systems. This paper describes the uses of that enabling technology with a focus on the practical application to existing heterogeneous C4I systems.

1. BACKGROUND

Information exchange between the current legacy of heterogeneous C4I systems continues to be a nagging problem within DoD in spite of the incredible advances in computing power and communications capability. Improving information exchange is not just a function of the state of information exchange technology, but also of the cost and practicality of implementation on a scale as large as worldwide C4I. This includes the systems of our NATO allies and coalition partners as well as US systems. Progress in providing effective information exchange in C4I systems must consider the cost of replacing that legacy with improved database and data interchange technology. Solutions to this shortfall are needed that will allow exiting systems to meet current operational requirements as well as provide a technology transition path over time.

The primary means for information exchange between heterogeneous systems in the existing C4I world involves the use of formatted messages. These are both character as well as bit oriented messages in a wide variety of types and formats. The U.S. Message Text Formatting program (USMTF) is the prime example. Two studies, one by the Institute for Defense Analysis [1] for DISA and one by the Army Science Board [2] for Department of the Army DISC4, provide insight into the magnitude, current status and possible future direction of formatted messages with a focus on USMTF. Existing message capabilities require: 1) expensive configuration management (USMTF alone has over 250 changes a year [1]); 2) a long time to implement changes (~2 Years for USMTF [1]); and 3) extensive software modification, testing and recertification [2]. Because of the time and cost associated with change, the users find ways of using the existing formats in ways they were not intended to be used. This includes use of multiple formats to transmit a few bytes of needed data at a high cost in bandwidth utilization, processing and storage [2], and use of free text fields to transmit entire formatted messages.

A program to overcome the shortfalls identified above by infusing new technology into the existing formatted message information exchange capabilities was begun in June of 1994. This program is under sponsorship by ARPA/ASTO with oversight from the Joint Staff J6 and the DISA /JIEO Center for Standards. The technical program, called the Data Transfer and Translation Module (DTTM), is a joint effort by the Georgia Institute of Technology and the Army Research Laboratory. Initial Proof-of-Concept of the DTTM uses one representative message format and two representative C4I systems as a basis for inference of the broader applicability of the technology.

The goals of the program are 1) to prove the feasibility of information exchange of data between C4I systems using self-describing message formats, generator-based parser technology and distributed data dictionary; 2) prove the feasibility of a general solution for information interchange between legacy and migration systems that use relational databases and receive data via formatted messages; 3) extend the technology into additional message formats; 4) provide an evolutionary path to transition the legacy of systems toward greater interoperability; and 5) transition the technology to the fielded systems. The first two goals have been met with the current implementation and the remaining three goals represent recently initiated work for future capability.

A Proof-of-Concept prototype for the core technology (goal 1.) was demonstrated in January 1995 [4] to include self-description, generator-based parsing, use of a shared database, and compliance with the Information Resource Dictionary System [6]. The ability to work with current legacy and migration systems (goal 2.) was demonstrated in July 1995 using two systems of the Global Command and Control System(GCCS): the Contingency Theater Automated Planning System (CTAPS) and the Joint Maritime Command Information System (JMCIS) using representative USMTF formats.

The architecture of the DTTM prototype is shown below:

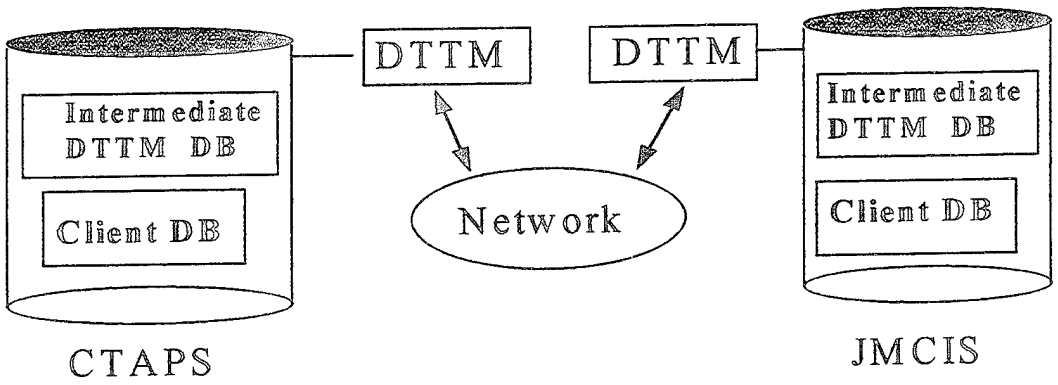


Figure 1: DTTM Proof-of-Concept Prototype Architecture

In [4] we described information exchange between DTTM databases. In this paper the focus is on the parts of the DTTM prototype that support the definition of the mapping between the DTTM database and the client databases, and also on the subsequent movement of data from the DTTM database to the client database.

2. DATA MAPPER AND DATA MOVER

A primary goal of the Proof-of-Concept prototype was to prove the feasibility of a general solution for information exchange between legacy and migration systems that use relational databases and receive data via formatted messages.

One key component of the technology is a tool which supports the definition of a mapping between the fields and sets in a message format and the attributes and relations in a client's relational database schema. A second component is a tool which generates a data mover. Once a mapping is defined for a particular message format between the intermediate DTTM database and a client database, all specified data elements in future messages of that type are automatically inserted into the client's database. This method of information exchange provides data integrity and minimizes the amount of data stored locally on a client's system.

The process used in the Proof-of-Concept prototype for moving data from one client C4I system to another is as follows:

1. A client uses the Data Mapper to map (or bind) parts of a USMTF message format for a particular message type to his relational database schema;
2. This mapping definition is stored within the DTTM system;
3. A Data Mover is automatically generated by the DTTM for that message type;
4. This Data Mover may then be invoked (either automatically when a message of that type arrives or via the query module) for any data message of that type;
5. The auditlog is updated with a date/time stamp after the data mover runs and again to reflect that the data was moved.

Several items should be noted at this point.

- one data mover generator exists for each USMTF message type, and typically legacy systems only deal with one to ten message types
- the client only gets the data he maps - not all message data
- a throw-away SQL script is generated by the Data Mover for each individual message
- the client has the option of manually moving the data or automating the process

Because of the self-describing nature of the DTTM database, USMTF message formats may be modified and new parsers automatically generated. That same technology allows the client message maps to be modified, and provides automatic generation of new data movers. The only restriction is that the fundamental meta-grammar for the USMTF standard cannot change [5]. Applying this technology to a new format (e.g. Variable Message Format, DIS Protocol Data Units) would require creation of another meta-grammar and new code in the parser generator and data mover generator.

General Solution

The concept of the DTTM was that of a general module that would be added to the environment of existing C4I legacy or migration systems that use a relational database management system and receives data from formatted messages. In the Proof-of-Concept prototype, the ability to update the databases of legacy and migration systems without modification to the current software was demonstrated. This required that we design the software to run within the current DoD standards.

Standards adhered to included ANSI C, ANSI SQL, and the Motif Style Guide. The DTTM Proof-of-Concept prototype was developed with the following tool set: Berkeley Unix k shell, gcc, bison, flex, Oracle V7, Pro*C, X Windows, and X Designer. The hardware environment consisted of a Sun Sparc 20 running Sun OS 4.1.3., HP 9000-720, and several Sun Sparc 5's.

3. DATA MAPPER GRAPHICAL USER INTERFACE (GUI)

The Data Mapper GUI was created to assist the end user in mapping his database to the intermediate DTTM database. The design goal was to define a tool that could be used by a data administrator or operator on site to update new and changed message formats in a matter of a few minutes or hours. The current procedure used in the USMTF configuration management system allows 12 months for updating the various fielded systems. This normally requires service or contractor modifications to the software. DTTM provides the ability to do the updates locally or to do them centrally and ship the changes out electronically. This allows for maintenance of strict configuration management while at the same time providing the capability for rapid message format updates to the system.

Global View

The global view of the DTTM Data Mapper window used for the Proof-of-Concept prototype is shown in Figure 2. The window is divided into four main sections: Message Selection Type, USMTF Message Format, Conversion, and Client Table Format. The minimal mapping of one field requires a message type to be selected, a USMTF set and field to be selected, and a client table and column to be selected. If data conversion or a default value is necessary, these are chosen last. To save the mapping into the intermediate database, the Confirm Mapping button must be selected. When all desired fields are mapped, the Generate option invokes the Data Mover generator to create a data mover executable for the particular message type. The Generate option also allows the client to set flags to control the process.

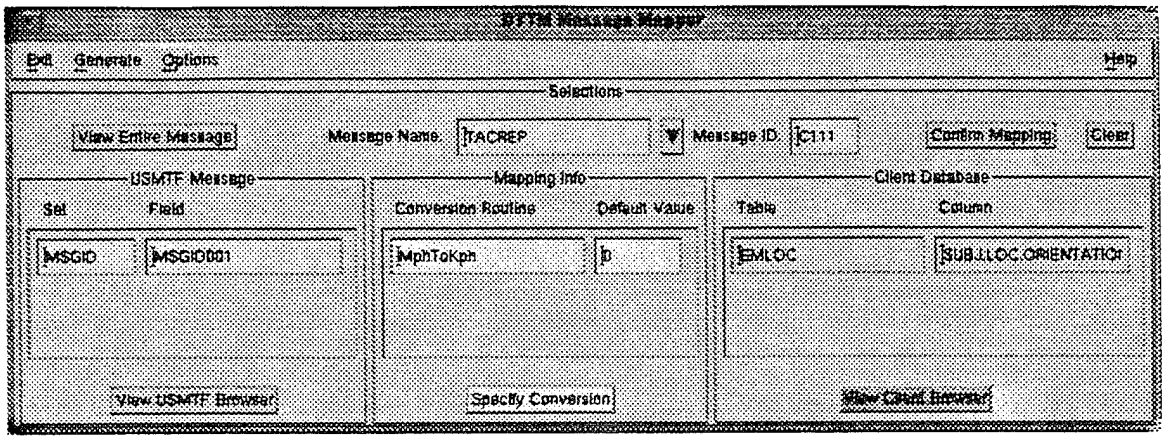


Figure 2: DTTM Message Mapper Window

USMTF Message Browser

The USMTF Message Browser window is used to display details about sets and fields belonging to the message type specified in the main mapper window. A diagram of this window is shown in Figure 3. The Set ID is selected from a list of all possible sets defined for the message type, and the list contains actual set names defined by the CDBS USMTF baseline. Once a set has been selected, the client may choose a field from a list of all possible fields defined for the chosen set. Additional data is displayed along with the set and field names to provide the user with all necessary information needed for mapping. The selectable push-buttons on the right side of the window provide an information box which describes the option in a free-text manner.

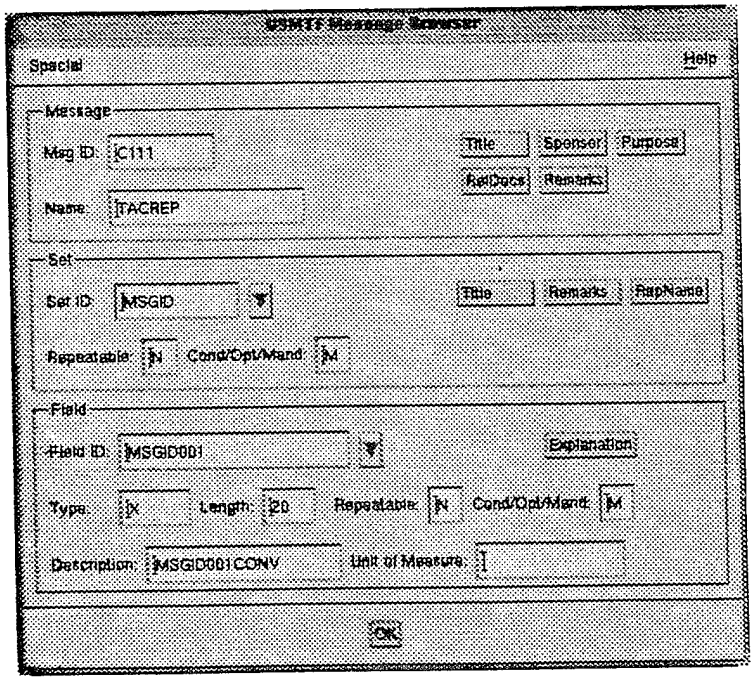


Figure 3: USMTF Message Browser Window

Client Browser

The Client Browser window is used to display table names and column names out of the client's relational database. A diagram of this window is shown in Figure 4. The table name must be chosen first from a list of client tables used to store message data. After the table is selected, the client can select a column from the list of that table's columns. Additional data is displayed along with the column name to provide the user with all necessary information needed for mapping.

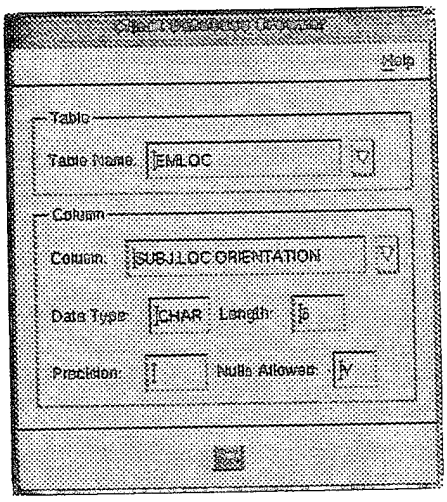


Figure 4: Client Browser Window

Conversion Window

The Conversion window is used if data needs to be converted from units in the DTTM system to different units in the client's system or if a default value is necessary. Data precision may also be specified for data conversions. A diagram of this window is shown in Figure 5.

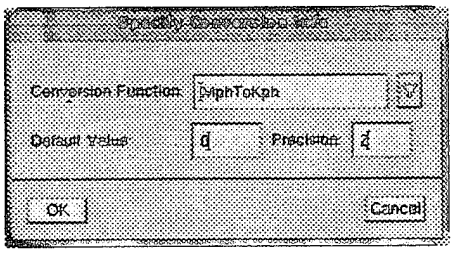


Figure 5: Conversion Window

4. ISSUES IN DATA MAPPING AND DATA MOVING

GUIs

Transitioning the technology from Proof-of-Concept prototype to fielded product brings up issues regarding usability and robustness in the hands of the target audience. Issues involved 'how to' do the following using the GUI:

- Make allowance for editing or deleting message maps.
- Make software that can meet a wide variety of environments and client systems.
- Provide checks and balances after a mapping is selected to ensure that correct data conversion is used when necessary.
- Previously-mapped fields should be designated so that the user can easily see how much of the message he is moving into his database.

Standard Data Units and Data Elements

The DTTM design concept included the need to utilize standard data units and standard data elements for internal representation. The idea was to incorporate the results of other research in data modeling and data standardization such as the C2 Core Data Model and the DISA Standard Data Elements. These were not integrated into the initial Proof-of-Concept prototype but would be added as the result of cooperative work with other DoD agencies.

One related design issue was how to handle data expressed in units. Our Proof-of-Concept prototype solution involved using a set of "standard data units" defined in the DTTM This was a necessity in order to design a category-based query module. It also simplified the data conversion process both in the parser and in the data mover. Both data values are stored in the DTTM database.

A more complicated issue is that of data conversion. The Proof-of-concept "standard data units" were used within the DTTM to evaluate methods of data conversion. It provides a set of routines which convert the incoming data into DTTM "standard" units and convert the outgoing data into client units as necessary. The client may also specify a default value for a field in case the message data for that field is null.

Mapping Granularity

Another design issue concerned the granularity of mapping. Should the client be forced to map all columns in one table before proceeding to the next? Should the client be allowed to map more than one message type at a time? Should he be forced to map one set at a time? Or, can he map any field of any message type in any order? The last variation was used in the Proof-of-Concept prototype with the restriction of staying within the same message type.

Mapping Types

One to-one mapping of data relationships was adopted for the proof-of-concept. A database schema relationships is defined with built-in constraints, primary keys, and foreign keys to enforce these. One-to-many and many-to-many data mappings will be added later to use the full capability of state of the art relational database management systems.

Primary Keys.

In several cases unique primary key values must be generated. Oracle provides a mechanism called stored procedures which may be called from a Pro*C program. A set of procedures which would return various values based on client input could be added as needed. Examples of return values are numbers generated from a sequence, letters randomly generated, a string of concatenated field names, or the input value itself.

"Many" Mappings and Repeatable Data.

Repeatable data is a subset of the many-to-one or many-to-many mapping types. The data mover generator would have to be extended to handle any type of multiple mapping. Examples of "many" mappings are listed below:

1. A DTTM repeatable field mapped to one client field (where multiple client rows are created for each instance of repeating data);
2. A DTTM repeatable field mapped to n client fields (where one client row is created, but at most n fields are populated with each instance of repeating data);
3. n DTTM fields are mapped to one client field (where one client row is created and the n DTTM fields are concatenated and stored in one client field);

Set and Segment Mapping.

The possibility of mapping entire sets or segments with one action was not included in the initial implementation. The relatively fixed mapping needed to do this would require standard data elements.

Version Control.

Currently, only one message map may exist for each message type; if a map is modified, it must be renamed. A version control mechanism which keeps track of message type and version number has been proposed to DISA. This would enable the DTTM to reconstruct messages from a previous version and store data for any version of any message type.

Data Validation and Data Units.

One of the hardest problems to solve is data validation and data conversion. Since USMTF message formats are extremely flexible, it is nearly impossible to convert all data to a standard

unit. One prominent example is the location field - it may be a Latitude/Longitude value, a UTM value, or a name like "Pacific Ocean". If the DTTM standard unit for location is lat/long, how do you convert "Pacific Ocean" to a lat/long value? Our Proof-of-Concept prototype solution was to store both the original data value and the converted value in case the conversion could not be done.

5. CONCLUSIONS

The DTTM prototype effort demonstrates the feasibility of information exchange between C4I systems using self-describing message formats, generator-based parser technology and distributed data dictionary; as well as the feasibility of a general solution for information interchange between legacy and migration systems that use relational databases and receive data via formatted messages. The path to realize the vision and provide successful technology transition lies in extending the software to address the issues described in this paper. The development team believes this approach is capable of achieving the objective.

6. REFERENCES

[1] J. Shea. "Assessment of the U.S. Message Text Formatting Program", Institute for Defense Analysis Paper P-2788, Technical Report for DISA/JIEO. Jan 1993.

[2]. I. Kameny and others. "Moving Army Tactical Command and Control System (ATCCS) from a Character-Oriented Message System to a Data-Oriented Message System", Army Science Board Issue Group Study, April 1994

[3] S. Chamberlain, Model-Based Battle command: A Paradigm Whose Time Has Come", 1995 Symposium on C2 Research and Technology: NDU, June 1995.

[4] J. Coleman, L. Mark , and A. Handy. " Data Exchange between Heterogeneous C4I Systems using USMTF". *The Proceedings of the 11th Annual DoD Database Colloquium*, August 1994.

[5] L. Mark and N. Roussopoulos. " Information Interchange Between Self-Describing Databases", *Information Systems, Vol. 15, No. 4,* (1990)

[6]. American National Standard Information Resource Dictionary System (IRDS, Proposed Draft), Parts 1-4, ANSI X3H4, New York (1985)

7. BIOGRAPHIES

James P. Coleman, Jr.
Senior Research Scientist, Georgia Tech Research Institute
Georgia Institute of Technology, Atlanta Ga. 30332

James Coleman is a Decision Scientist and Program Manager on software development efforts for military C4I programs. He successfully managed a multi-year development program on over 20 separate contracts for software components of Air Force Mission Support Systems (MSSII, MSSIIA, and AFMSS). Other ongoing research includes database navigation and visualization software for C4I systems using Hypermedia technology, for Rome Laboratory. Earlier programs he has managed include a decision support planning aid for the Army's Combat Service Support Control System, a "Guide for Decision Support System Development" for the Army's Information Systems Command, and a methodology for threat data analysis on the Heuristic Route Planning Optimization (HERO) program for Rome Laboratory. Mr. Coleman is a former Air Force Officer and member of the Georgia Air National Guard. He is an active member of AFCEA and is on the Board of Directors for the Atlanta Chapter (404) 894-8959, jim.coleman@gtri.gatech.edu

Lisa C. Sills
Research Scientist II, Georgia Tech Research Institute
Georgia Institute of Technology, Atlanta Ga. 30332

Lisa Sills is a Research Scientist specializing in database technology and system integration at GTRI. She has been involved with several database applications including accounts receivables, educational records processing, and government billing systems. She has also led many technical teams on multi-million dollar contracts for multi-year programs. Ms. Sills was employed by Unisys and Control Data Corporation before joining GTRI, and she is an active member of ACM and AFCEA. (404) 894-8957 lisa.sills@gtri.gatech.edu

Building the Infrastructure for Client/Server Applications

Barbara Timblin
Client/Server Specialist
Symantec Corporation

Introduction

Client/server computing is more than building pleasing interfaces to remote data. The focus of this paper is on what supports those pleasing interfaces: building the infrastructure for client/server applications. With the proper infrastructure client/server computing can support the larger context of a full-scale, enterprise-wide computing environment.

In most companies, support for complex, integrated applications spanning multiple computing architectures and expansive wide-area and local area networks will needed. And since these organizations are dynamic, rapid change to these systems will be required. Critical business systems for a global organization will increase the complexity by orders of magnitude. In addition, as the world embraces this changing market, long cycle times will not be tolerated. Business will be re-engineered.

The Environment

In this changing environment, the corporation must be recognized as the real "client." Client/server must serve the corporation, *not* just the individual user. The problems and opportunities in this environment are indeed big -- in fact, they are *huge*.

Despite major advances in both hardware and software technology over the past 30 years, the fact of the matter is that the *basic* requirements have not changed much. Sure, users want more sophisticated interfaces and, as more and more of them become computer literate, they desire more control over their environments, but the basic need for timely, accurate information hasn't changed.

We're only now getting good at satisfying these requirements in a mainframe environment. Client/server computing is a whole new game. Problems will magnify as applications are spread across multiple computing architectures;

different locations spanning space, time, languages, etc.; complex networks; business pressures demanding more for less; and a much less technical user base.

Effective, enterprise-wide client/server computing is a multi-year, evolutionary process, not just a current "point in time" event. To be successful, client/server computing must strike a careful balance between the rules and rigors of MIS and the demands for freedom by end users. And then all of this needs to be synchronized for successful implementation

Complexity of Interaction

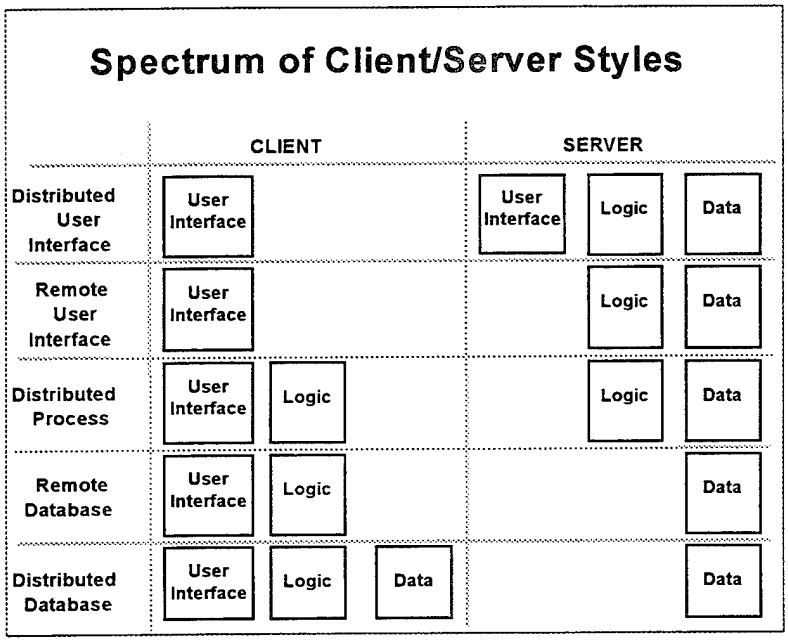
Client/server computing is a step along a natural evolutionary path towards business-driven development that has been enabled by dramatic advances in hardware technology.

You can begin to classify client/server applications based on *complexity of interaction*. From a simple client/server environment -clients all accessing one database -a homogeneous architecture; to a tiered client/server environment with multiple databases and finally to the enterprise-oriented client/server infrastructure. Enterprise client/server involves intelligent workstations interacting with a variety of servers on a massive network. Work can be offloaded, rerouted, and generally spent speeding to a number of cooperating servers (some of which may also be clients) on the network.

Please note that the mainframe is still included. For most organizations, the mainframe will continue to be an important, integral part of the client/server environment facilitating massive data sharing across the enterprise. Its function, however, will change drastically. Instead of being the center of the IS universe, the mainframe will play the role of server -- granted, a very powerful mega-server -- cooperatively interacting with smaller processors and intelligent workstations.

The world of monolithic computing had its advantages, mature infrastructure, well-understood principles to ensure maximum reliability with optimum performance. Users who are enamored of client/server computing today will become less so when they feel the absence of the infrastructure elements upon which they have come to rely so heavily.

From the developer's perspective, client/server computing is all about where the presentation logic, business logic, and data manipulation logic are placed. There is a wide array of alternatives, including Distributed User Interface; Remote User Interface; Distributed Process; Remote Database; and Distributed Database. Or any combination of these styles.



It is important to find tools that can support this spectrum of application styles. For enterprise-wide client/server computing, an organization will require many if not all of these alternatives. For example, the "Distributed Process" option is essential for any serious effort at client/server. It is important, then, to find tools that impose few or no restrictions on distribution.

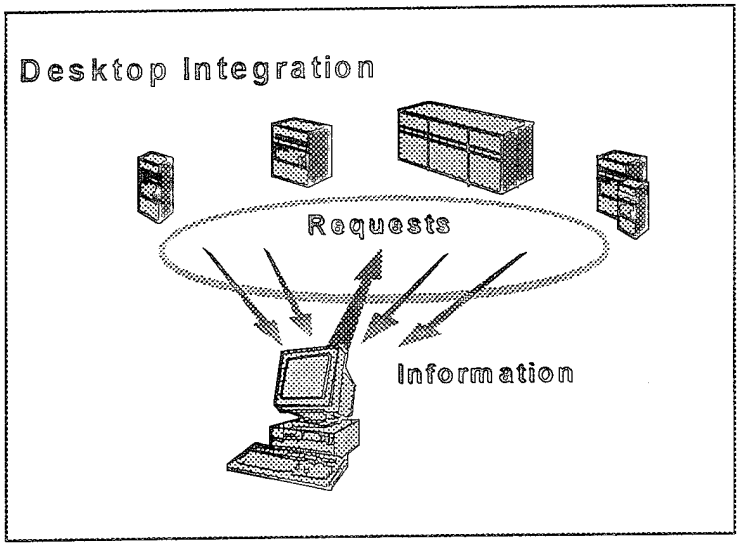
The User

The user has a different view. To the user, the term "Client/Server" does not represent a hardware configuration. Despite the fact that there are many interpretations with differing opinions about what the term really means, from a user's perspective there is a simple conceptual definition. The user's intelligent workstation is a client. The rest of the computing environment to which the client is attached and with which it interfaces is the server. The server is likely to be made up of a great many interconnected machines, but to the user it appears as a single "virtual computer" dedicated to satisfying his or her needs.

Additional distinctions (such as peer-to-peer, requester/respondent interactions, cooperative processing in general) might be interesting, but client/server is the definition of interaction from the perspective of the end user of the technology.

One trend we're seeing today is that more and more of the user's interface to application is becoming configurable. That is, one aspect of empowering users is that they can change the interface to the applications they use.

Besides providing users some gratifying flexibility, this actually allows them the ability to address new requirements as they are able to *combine* applications with more ease.



Today's power-users are interested in the flexibility of desktop integration. This is one of the prime benefits of client/server computing: the client can bring together a variety of applications and tie them together.

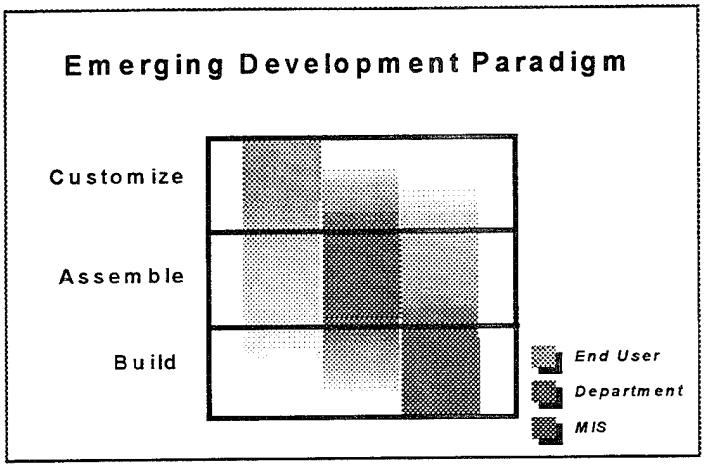
Desktop integration might at first seem an expensive luxury or, worse, a toy. However, it provides the basis for *process* integration. That is, in order to tie together the components of business processes that might be scattered across many heterogeneous machines running apparently incompatible software, the desktop can provide the glue. Approaches like Microsoft's OLE provide a means by which previously isolated application components can finally communicate with one another via the desktop.

Shift in Responsibility

This trend is leading to a new approach to development. In the past, MIS was solely responsible for the creation of application systems. Now, however, we're seeing an increasing shift in responsibility that is leading eventually to a situation in which:

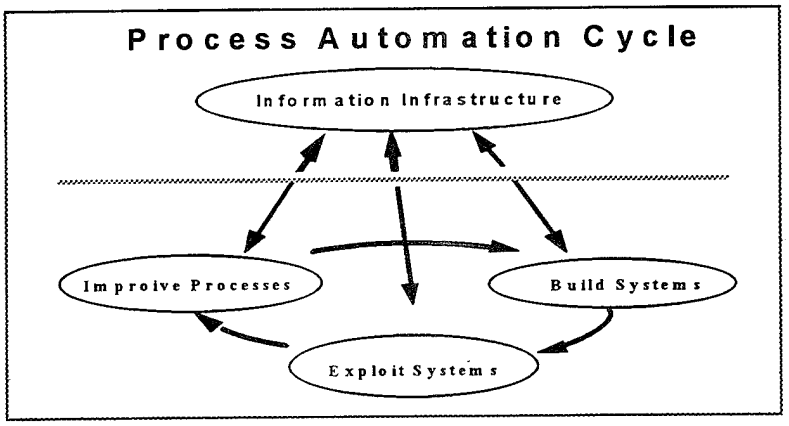
MIS is responsible for building reusable business objects. These objects are the basis for providing cross-functional process support.

- Individual departments can assemble these objects into meaningful applications supporting a meaningful segment of the process.
- The end-user is empowered to customize the application to best meet his or her specific needs.



Thus, from the perspective of the technologist, a mechanism must be found that supports the distribution of enterprise data and objects across the enterprise and throughout components of the enterprise. Users must be given the tools to create their own objects and to tie them together by means of the features in their user interface.

Now let us stop for a moment and look at the Process Automation Cycle. As organizations use this cycle (improve a process; build a system, exploit the system), they can begin to develop information assets in the form of *business objects* as part of their information infrastructure. Thus, the automation of business processes results in the emergence of potentially reusable components. That is the meaning of the arrows pointing upwards.



To gain any benefit, though, these assets must be reused; that's why the arrows also point downwards. An organization that can take advantage of work by reusing business objects will be able to accelerate the process automation cycle greatly.

To do any serious client/server development, a number of critical requirements must be met: to implement the process automation cycle and provide a structure for asset development and reuse.

Strategies for Implementation

Point tools or First Generation Client/Server development tools are well suited to the construction of simple client/server applications. Yet, the most popular of the point tools do not enable component sharing and application consistency by means of a shared repository. They do not support transaction processing. They do not isolate business rules from implementation details. Most importantly, their approach to client/server computing focuses on a very small band of the client/server spectrum: the accessing of data remotely. They have no facilities for distributing the *logic* of an application process across multiple machines.

These features *are* included in the best of the Next Generation tools.

The spectrum of application complexity is become more broad over time, from simple ad hoc queries; to more complex queries requiring access to multiple data sources; to simple update processes; to multiple database update processes which may have to retain conceptual integrity between databases housed on multiple heterogeneous platforms under the control of multiple DBMS's; to applications that breach the 'enterprise barrier' and interact with external organizations (through EDI, perhaps.) The development environment must be rich enough to handle this rich array of computing problems with which it is now forced to deal.

The underlying business rules change at a different rate than the implementation technology. Thus, you can minimize your investment in technology upgrades by using model-based development to isolate the business rules you've worked so hard to discover from the implementation details wrapped around them.

Powerful enterprise applications require a solid infrastructure to ensure integrity and reliability. An environment that truly supports business-driven development.

Business-Model Development

The term "enterprise engineering" refers to a series of activities associated with ensuring that the enterprise's requirement for information is met in both the large (with globally shared information) and in the small (with local services). It has been defined as: "A comprehensive approach to designing business processes and automating them within an enterprise-wide infrastructure."

No client/server implementation can be successful if the problem at hand is not well-understood. And there is no better substitute for understanding a problem

than the generation of a well-designed and consistent business model including all of the rules implicit in the processes of that model. Managers of many enterprises acknowledge that business processes today are simply too complicated to understand without constructing models of the data and the processes that drive that data. Models become the means by which large, complex problems are abstracted and thereby understood.

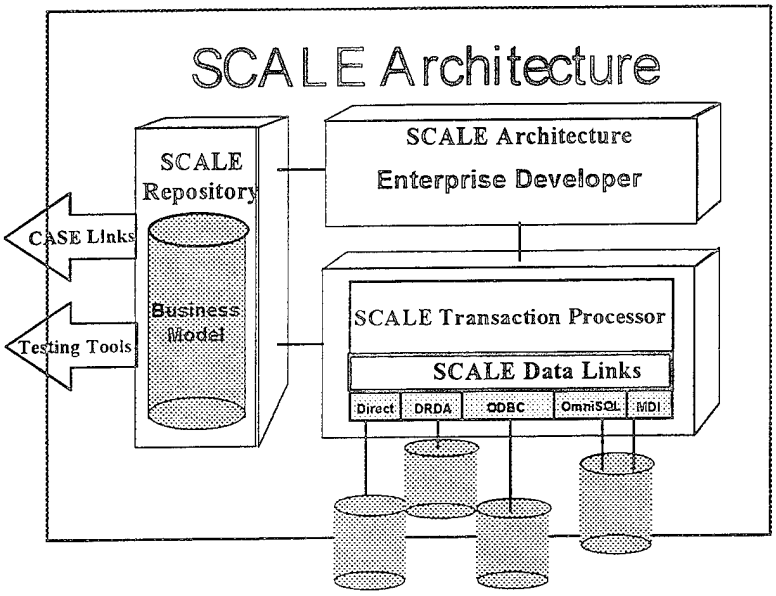
Beyond obtaining a better understanding of a problem, business models offer other benefits for software application development:

- The ability to update and maintain an application is vastly simplified when the business model and rules are isolated from the physical implementation of the application.
- By creating a single point of reference, it becomes easier to identify inconsistencies and incomplete processes.
- Upsizing, downsizing, and brand-shifting of the physical infrastructure are isolated from applications and become independent of the business processes.
- Understanding the impact of changes in the business environment does not require knowing the implementation details of an application.
- Logical abstractions of the business model isolate it from specific hardware implementations or deployment strategies.

A centralized repository or business model automates the processing of client-server transactions optimized to each data source. It isolates the enterprisewide logical datamodel in a central active repository. This repository also contains the data road map, which includes the types of data engine and the location of the data. It also stores centralized business rules, which are defined in a declarative manner and complete the business model. In essence, by leveraging information in the business model, transaction processing is automated and access to relational database management systems are optimized.

Client/Server Architecture and Tools

This next-generation client-server architecture, that isolates and automates the client-server interaction based on a business model, is implemented with Enterprise Developer. This architecture provides for very rapid application development, including applications that are scalable form standalone to enterprisewide applications.



Business Model. SCALE centralizes the business model in a repository and automates the processing of client/server transactions optimized to each data source. SCALE isolates the enterprisewide logical data model in a central active repository. This repository also contains the data road map, which includes the type of data engine and the location of the data. It also stores centralized business rules, which are defined in a declarative manner and complete the business model.

The repository reverse-engineers existing databases, creating an entity relationship model. New entities can be created in the repository and then generated into physical tables. Repository information represented in the SCALE business model is automatically incorporated in Enterprise Developer applications. Alternatively, when Enterprise Developer operates a s a rapid prototyping tool, SCALE automatically captures the business model in the background.

In essence, by leveraging information in the business model, SCALE automates traction processing and optimizes access to relational database management systems.

SCALE Transaction Processor. The SCALE transaction processor uses this model to isolate and automate the processing of client-server transactions. SCALE furnishes an innovative transaction processor to automate optimized processing of client/server applications. The SCALE transaction processor features:

- Multilevel master detail transactions
- Automatic enforcement of business rules

- Optimum retrieval processing
  - caching of retrieved data
  - query partitioning
  - SQL generation
- Optimized commit processing
  - Optimistic concurrency
  - Two-phase commit for distributed updates
- Heterogeneous distributed database support

SCALE DataLinks for Optimized Data Access. SCALE provides direct connections to mission-critical data sources, generating optimized SQL Statements fine-tuned to each data engine. These fine-tuned optimized connections take advantage of engine-specific features. SCALE can also uses gateways such as ODBC, Sybase's OmniSQL Server, and Micro DecisionWare Gateway to access other sources of data.

SCALE enables rapid development of client/server application allows them to access heterogeneous distributed databases, makes them Scalable to changing data configurations, centrally applies business rules, and supports client/server life cycle development methodology through isolation of visual and data functions.

Based on SCALE, Enterprise Developer is a visual object-oriented application development tool. Its "programming by exception" model allows complex forms to be developed within minutes. These forms can be extended and customized to build powerful applications using the visual editors, the built-in 4GL called SCALEScript, and external objects and programs using its open object architecture. It provides a unique integrated environment for developing data entry forms and reports, as well as on-line decision support systems.

In Summary

The process automation cycle (improve a process; build a system; exploit the system) here is the basis for enterprise client/server computing as follows:

- Improved business processes yield new applications that are more effective
- The information systems built will now have components scattered across multiple machines

The exploitation of client/server systems is essential to accomplishing the business processes.

Remember there are three major components to the migrating to an enterprise wide client/server computing environment.

1. To Implement the Technical Infrastructure you must:
    - Implement the Network
    - Proliferate Workstations
    - Deploy Servers
    - Implement Network Management Tools
    - ...and Implement Network Security

2. Then business-driven development standards must be established. They include:
    - Instituting a Development Methodology
    - Setting Architectural Guidelines
    - Determining Transition Strategy
    - ...and implementing an architecture-independent environment

3. Finally, you can begin to build applications and refine the environment by:
    - Benchmarking Best Practice
    - Leveraging Remote Database Architecture
    - Executing the Migration Strategy
    - Reviewing and Adapting to Changing Business Drivers
    - Reviewing and Adapting to Changing Technology Platforms

The chief message is this: you *must* begin to move ahead along this path, but you must do so carefully.

Barbara Timblin
Client/Server Specialist
Symantec Corporation

Ms Timblin has had extentsive Client/Server experience. As a consultant for Oracle she implemented a worldwide client/server application. In addition, she supported the case tool from Texas Instrument, Information Engineering Facility (IEF). Ms. Timblin has a two degrees in Computer Science; Bachlors from Manhattan College, Riverdale NY and a Masters Degree from Drexel University, Philadelphia PA.

Using Automated Workflow Systems and the Internet to Manage Corporate Data Standardization

Bonnie L. McHenry, Peter J. Magee

Abstract

One of the biggest problems that any major corporation faces is controlling the proliferation and duplication of data systems. To control this problem, a corporate data dictionary as well as data standardization procedures should be established. An organization, such as DoD, can implement a data dictionary system that controls both data and management of that data. This data dictionary system and data standardization principles can be of immediate benefit not only to corporate management but to the functional users as well.

Since many organizations are trying to deal with old systems developed prior to the widespread acceptance of data standardization principles, in addition to new systems developed under more stringent guidelines, an initial approach to controlling data proliferation and redundancy is to standardize the existing interfaces between systems. Once these interfaces are tracked more efficiently it becomes easier to begin the data standardization process for all data elements.

In the case of DoD, a corporate dictionary of system interfaces can make use of existing resources like DIST and DDRS, which document data systems and data elements respectively, to show how data flows between systems, who originates the data, and who the end users of the data are. The interface dictionary can be designed in such a way that it not only provides high level reports for corporate planning, but also provides functional users and system administrators a workflow environment that makes the job of documenting interfaces simpler and more efficient. In addition, the Internet can be used to provide an environment that allows corporation-wide access to dictionary.

1.0 Introduction

Full scale data administration and standardization is one of the most important and monumental undertakings that the DoD data systems community has ever undertaken. It holds the promise of allowing corporate management to make more efficient and economical use of its data resources, helping system administrators to develop new and better applications and to perform more accurate impact analysis on system modifications than ever before. Unfortunately, the concepts of data standardization are not always understood or supported by the systems community.

Data dictionaries, which have the potential to be useful tools for all levels and types of users, are often developed solely to provide high level reporting and analysis to upper level management and data administration. We would like to propose that functional users, who provide the lion's share of the data that goes into a data dictionary, can benefit just as well from a corporate data dictionary. If functional users are included in the design process and are provided with the proper application software, all users can make better use of their data resources; therefore, increasing daily productivity and efficiency.

2.0 Background

The Department of Defense, like many large and diverse organizations, has a history of developing independent "stove pipe" data systems in isolation. Because of limited funding and personnel little research time is spent on making more efficient use of the existing resources. This situation is often aggravated by rivalry and competition between different organizations and services with similar responsibilities. For example, the Army may a design stock control system one way; however, the Navy may have their own stock control system that does same job but the design it completely different. As a result a great deal of redundancy, both of data and functionality, has developed among DoD systems.

Because these independent systems often need to share data, a large network of interfaces has developed over the years to keep the independent systems synchronized. Often these interfaces are insufficiently documented, and it is difficult for system administrators to determine the original source of incoming data (if there is a single authoritative source), or to be fully aware of the consequences to other systems of reformatting outgoing data. The validity of much of the interface data becomes suspect when its source is unknown or not properly documented.

Currently, the Defense Information Services Agency (DISA) is sponsoring two projects that contain the potential roots of future DoD data standardization and administration. The Defense Information Services Tool (DIST) is a dictionary of DoD data systems, and the Defense Data Repository System (DDRS) is a dictionary of data elements. However, there isn't a DoD corporate level dictionary of system interfaces to tie these two resources together.

3.0 Goals

Our goal is to establish a DoD corporate interface data dictionary, to standardize all interface data elements, and to improve documentation of all system interfaces. We propose that this be accomplished through the implementation of efficient, workflow-oriented software tools that make full use of the power of client-server architecture and the Internet. Standardization of interfaces will allow upper level management to determine they type of information being shared between existing systems and to identify places for data or functionality consolidation. By linking the interface dictionary with other tools, like DIST and DDRS, it will be possible for system administrators to identify more accurately the sources of incoming data and the potential sources for new data, and to perform detailed impact analysis of system modifications. As the task of documentation is made more efficient by workflow-oriented software, accuracy of data within the dictionary will improve, making it easier to begin the data standardization process for all non-interface data elements.

4.0 Approach

While their potential is undeniable, there are some limitations to tools like DIST and DDRS in their current configuration. Both tools have been designed principally as corporate level reporting and analysis systems, with little direct visibility to functional users in the data systems community. Other systems and more detailed dictionaries, which provide metadata to these tools, get no direct benefits from the collection and maintenance of this metadata once the information is provided. Participation, when it occurs, is generally unenthusiastic. Many systems have been affected by the downsizing of personnel and budget allowances that plague most of DoD. Administrators of these systems are unlikely to spend time and money on projects that have no direct benefit to them. Therefore, one of the objectives of corporate data administration should be to create data dictionaries that not only can provide all of the planning and analysis reports that are needed by upper level management, but also provide quantifiable benefits to the entire DoD data systems community.

The first way to provide direct benefits to the functional users and system administrators in a large corporation is simply to provide direct access to the data dictionary. According to Dureil (1985), "one of the most important benefits of data administration is to share metadata with the user community" (p. 17). Distributed processing, personal computers, report writers, query languages, and networks, including the Internet, have all been developed to allow greater, more flexible access to centralized data resources. Many corporations, including DoD, have already made these kinds of tools available to a large percentage of the user community, but without access to the data dictionary as well as their specific data systems these tools are of limited value.

Functional users need access to their specific data systems; however, they must also have access to information (or metadata) related to their systems and the other data systems with which they interface. Access to corporate metadata allows users to research characteristics of other data resources in the corporation, and perform detailed impact and design analyses on their own systems. Greater access by users to the data dictionary gives "data administration the opportunity to actively contribute to the improvement in design and usage of new data structures and systems" (Durell, 1985, p. 136). These features can have a measurable impact on how functional users perform their jobs. If user's jobs can be positively impacted, using tools they already possess, they are better able to see the benefits of data administration and standardization and participate enthusiastically in such projects.

A second way to provide direct benefits to functional users, as well as to data administration personnel, is to give them a way to maintain part of the metadata in the dictionary. This can be accomplished by creating a set of workflow-oriented applications that help users get their day-to-day jobs done faster and more efficiently, while aiding the higher goals of data standardization and corporate data management. "Workflow software provides productivity gains through the automation of paper tasks. Electronic forms travel more quickly and are easier to store and retrieve than their paper equivalents" ("Intro. to Workflow," 1994).

For example, the formal documentation of system interfaces can be an extremely complex, drawn out process, involving multiple documents and reference materials. By using existing client-server technology, it is possible to create applications that not only automate the data entry and coordination of the interface document in the data dictionary, but also aid system developers in interface design by providing access to data element definitions and record structures. These applications can simplify and reduce the workload for both the functional users and data administration personnel, who approve the interface documents, by reducing the amount of data entry and research required to generate an interface document. This allows interfaces to be documented and coordinated more quickly and accurately than ever before. Thus metadata, such as interface documentation, can be made more accessible and more meaningful to users of the dictionary at all levels.

5.0 Case Study

Data administration and data standardization offices in the Department of Defense have had a general lack of success in persuading functional users and data system administrators to provide accurate information in a timely manner ("What's in it for me?"). As in many large corporations, the information on older "legacy" systems, interfaces, and data elements necessary to build a comprehensive data dictionary has never been adequately documented. Traditionally, no documentation was generated until the systems were implemented, and then it was done quickly, while developers could still remember how the system worked, and incompletely (Durell, 1985, p.22). However, there are a few dictionaries that have been able to maintain a relatively high level of accuracy over the

years either, including the Data Standardization Office (DSO) of the Material Systems Group, which is part of the Air Force Material Command.

DSO (currently, AFMC MSG/END) has spent several years documenting interfaces between AFMC legacy data systems. During this time accuracy of the interface documentation was improved with annual reviews and reconciliations. Fortunately, DSO had a relatively comprehensive information base and cooperative users. However, Interface Control Documents (ICDs), also known as Memoranda of Agreement (MOAs), existed largely only on paper. Only two of fourteen paragraphs or sections on each ICD were entered in the dictionary; the complete ICD existed only on paper. Generation and approval of an ICD could take as long as six months while the paper document and its attachments were mailed from desk to desk, waiting to be signed by the functional users and network managers and finally approved by DSO.

DSO, like almost every other organization in DoD, began to feel the budget crunch. As monetary and personnel resources dwindled the workload within DSO became more and more backlogged. However, the number of interfaces and data elements to be documented rose because of initiatives like DIST and DDRS; the workload steadily increased. Projects were delayed or shelved entirely as DSO personnel attempted to cover the highest priority items in the backlog. A major change in the way DSO performed its function was required. What resulted has had more impact on DSO in terms of its image within the command and with its customer base (MSG legacy data systems) than almost any other single action.

It was determined that a way was needed to speed up the ICD coordination process, and to place as much of the ICD development responsibility with DSO's customer base, which is where the responsibility belongs. The ICD portion of the dictionary was expanded to include all sections of the paper ICD. Once the paper ICD was eliminated, the official ICD would exist only in electronic form within the dictionary itself. Functional users would be able to enter draft ICDs directly into the system, instead of filling out a paper form, and a large portion amount of the data entry workload was removed from DSO without putting any significant new requirements on the customer base.

A workflow Windows application was then designed around the draft ICD, controlling its coordination cycle. According to Coulouris, Dollimore, and Kingdberg (1994), "designers of systems must consider the needs of potential users" when developing a workflow application (p. 49). For the Interface Control Document System, developers worked closely with DSO personnel and the functional users of the new system to accurately model the business processes and rules associated with interface documentation. An effort was made to mimic electronically what is currently was being done on paper as closely and intuitively as possible. This type of contact with the users of the software is essential; no one understands how a person does their job better than that person. A workflow application developed in a vacuum, isolated from user feedback, is almost certainly doomed to receive poor response from its intended users. In addition,

part of the work process will be misunderstood, misinterpreted, or badly designed by the application developers. The end users are unlikely to use the product, especially if the resulting system is not intuitive or makes their work more difficult.

With these principles in mind, the developers developed the application to simulate the functional users procedures in an automated environment. For example, as a draft ICD completes predefined milestones on its path towards completion and final approval, the application automatically detects its status and generates e-mail messages to the functional users, network managers and DSO who are on the coordination list, keeping everyone abreast of the ICD status. During coordination, functional users and points of contact for each system involved in the interface can access the system, see the electronic ICD, and indicate their approval or disapproval at the press of a button. Electronic signatures (userids) are registered in a coordination history. With this new methodology, an ICD can go from initiation to final approval in days instead of months, enabling functional users to generate approved interfaces in record time and DSO personnel to document more interfaces then ever, even with reduced personnel.

By designing the application software to connect to the core DBMS over the Internet, the DSO data dictionary is available from literally anywhere in the world. At demonstrations of the system, the revised dictionary gets enthusiastic response from DSO's original customer base. In addition, administrators of data systems, dictionaries, and repositories from other commands and groups are interested in adapting the new software or merging their data to form a more comprehensive dictionary.

6.0 Workflow

The heart of the software designed and used by DSO is the ICD workflow cycle that aids functional users in the creation and proper documentation of their interfaces. The workflow cycle, more than any other single aspect of the system, is what makes the new client-server dictionary a success story, and it is this type of application that can bring the most benefits to the corporate data systems community.

As the user begins working on a draft ICD, their work is checked at each step of the data entry process. Before the user can even begin entering details of an interface, the application verifies that the systems involved are already registered in the dictionary. Screens are designed to break the process of data entry into logical units, enabling the user to enter the ICD data in an intuitive, step by step process that requires a minimum amount of training. Reports are available to tell the user what stage the ICD is in at any time, or if critical information has been left out of the ICD. As users log into the application to coordinate an ICD, they are prompted automatically, based on who they are and what their role is in relationship to the ICD, to take appropriate actions, or they are presented with a view-only screen if no action is allowed. Comprehensive, context sensitive help is available at all times.

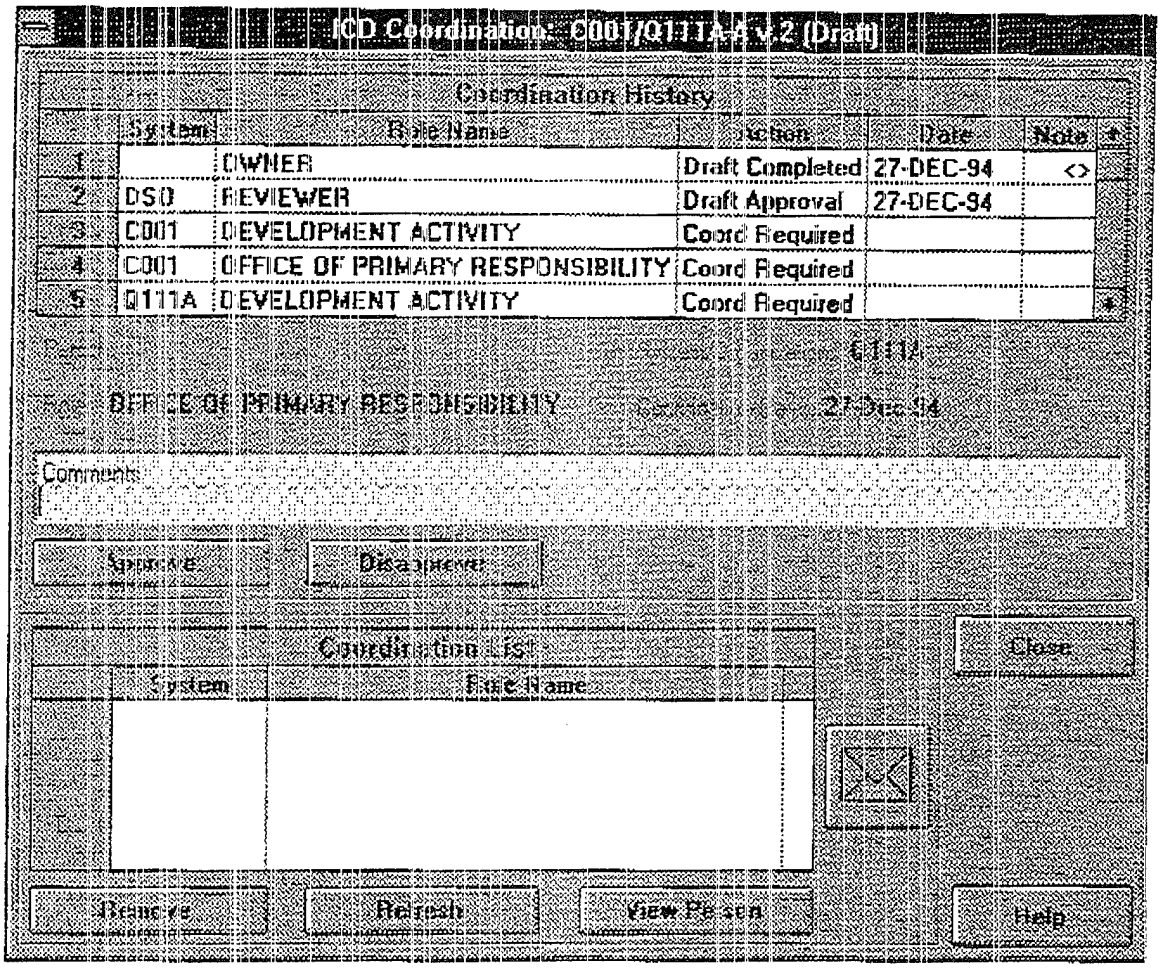


Figure 1: Sample ICD Coordination Screen

In addition, actions by users can alter the course of the ICD design process (see Figure 2). If a disapproval is registered during coordination of the ICD, e-mail is generated notifying the persons on the coordination list of the action and the justification for that action. The application prevents the ICD from receiving a final approval until the situation is remedied to the satisfaction of everyone on the coordination list. Subsequent rounds of coordination are required until all users have approved the ICD. At that time, DSO grants final approval of the ICD and the process is complete.

The electronic format of the ICD and the automation of the coordination cycle remove most of the slack time from the coordination process. Instead of sitting idle for days or weeks at a time, buried in somebody's "in basket", the status of the ICD can be monitored at all times. E-mail messages and other notifications keep the ICD moving through the cycle; bottlenecks in the process are more easily identified and eliminated by data administration. And the user's work becomes more accurate, more timely, and easier to complete.

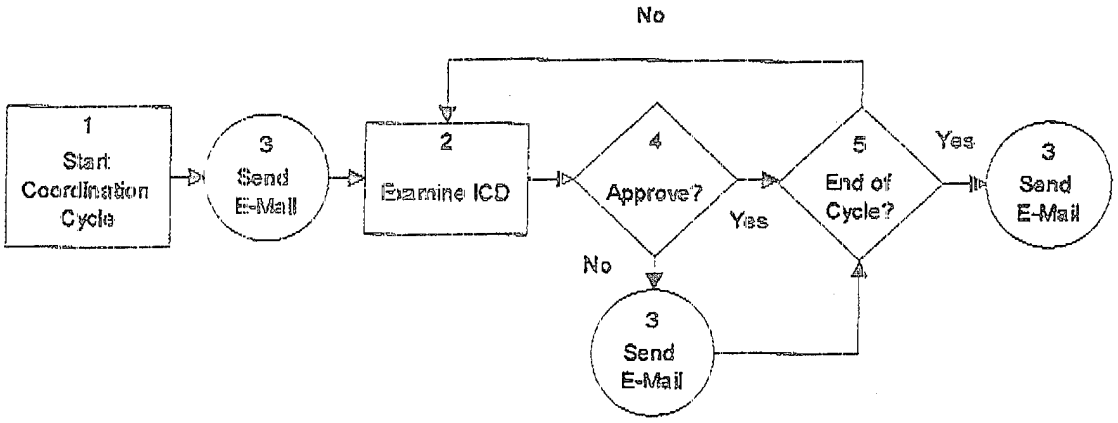


Figure 2: Simplified Coordination Process

The ICD workflow cycle incorporates both of the key aspects discussed earlier in this paper. It gives users access to all the data required to construct an ICD, and guides them through that construction as much as possible. It monitors completion status, and at specific points in the coordination process it prompts the parties concerned with the ICD development to take critical actions. Metadata becomes more accurate and meaningful, and day to day work activities for both functional users and DSO are made more efficient.

7.0 Internet

The Internet is literally the glue that binds the workflow system together in a global working environment. The Internet enables computers to communicate with one another in a uniform manner, despite location, local network protocols, or workstation operating systems. This means that users across an entire world-wide corporation, like DoD, can get access to centralized data resources, often without needing to make any expensive upgrades to their existing hardware. Software (like TCP/IP) required to access the Internet is generally inexpensive and easy to find.

This Internet access ensures that not only does everybody who needs access to the system have it available at their own workstation, but it ensures everyone is looking at the same data all the time. This eliminates problems that have occurred in the past with distributed system, where there is a constant struggle to keep the data synchronized.

8.0 Conclusion

Data administration and data standardization are necessary to reduce duplication of data and functionality among corporate data systems, and data dictionaries are key tools in that effort. Unfortunately, more attention is often paid to the capabilities that such tools give corporate management in terms of resource analysis and reporting than to the equally important capabilities those tools can give to the general user community. When given

proper access to a data dictionary and the right types workflow-oriented software tools, the lives of both data administration personnel and the average functional user can be greatly improved.

It is important for data administration to remain in close contact with the user community. Data administration requires their support to maintain accurate and up-to-date metadata. That support can be gained and fostered by providing users with tools that assist them in their day-to-day activities, as well as promote data administration's goals and objectives. Therefore, it is important to carefully model the processes and tasks that users perform when designing workflow software. End users should be included in every step of the design process.

The Internet provides a excellent medium through which a geographically distributed corporation, like DoD, can implement corporate-wide data standardization in the form of data dictionaries.

References

An Introduction to Workflow, (1994). World Wide Web at gopher://gopher.mit.edu:3714/0D%20tiserve.mit.edu%209000%2038541%20ready.

Coulouris, George, Jean Dollimore, and Tim Kindberg, (1994). Distributed Systems: Concepts and Design. London: Addison-Wesley Publishing Company.

Durell, William, (1985). Data Administration: A Practical Guide to Successful Data Management. New York: McGraw-Hill Book Company.

Peter J. Magee
Senior Programmer/Analyst
NCI Information Systems, Inc.
3146 Presidential Drive
Fairborn, OH 45324
Office: (513) 427-0252
Site: (513) 257-6670 or (DSN) 787-6670
E-Mail: mageep(@wpdis01.wpafb.af.mil

Mr. Magee graduated from Purdue University in 1990 with a degree in Aeronautical and Aerospace Engineering. His many hats to date have included technical writer, contract management, fluid systems engineer, systems analyst, programmer, data modeler, data base administrator, software trainer, and general repository of all things technical. For the past three years he has been a member of the design team responsible for the development of the interface dictionary described in the case study, and is currently serving as its technical lead.

Bonnie L. McHenry
NCI Information Systems, Inc.
3146 Presidential Drive
Fairborn, OH 45324
Office: (513) 427-0252
Site: (513) 257-6670 or (DSN) 787-6670
E-Mail: mchenrb(@wpdis01.wpafb.af.mil

Ms. McHenry graduated from Wilmington College in 1977 with a major in English and a minor in Computer Science, and obtained her master's degree in 1994 in Composition and Rhetoric. She has spent her professional career in the computer industry as a programmer, programmer/analyst, systems analyst and project manager. For the past year, she has been the project manager for the design team responsible for the development of the interface directory described in the case study.

MIGRATING TO OPEN SYSTEMS AND RDBMS TECHNOLOGY

DAVID THOMPSON
ACUCOBOL, INC.

1. MIGRATION

Acucobol is dedicated to producing products that provide COBOL users with modern Open Systems tools.

Agenda

1) What is Rightsizing, and why are we considering the move to Open Systems

2) What are the main Rightsizing Alternatives

3) Legacy COBOL Migration Methods and Tools

4) Implementation and Personnel Issues

5) Case Studies

What is Rightsizing

Rightsizing is updating an older mainframe or proprietary mini-computer application (a COBOL application) to take advantage of the modern Open Systems hardware and software environments.

What is driving this movement to Open Systems?

Yesterday's Computer

Digital computers have been around since the end of World War II.

ENIAC – The first modern computer

Today's PC

The modern PC has processing power and data storage capabilities that could not have been dreamed of in ENIAC days. The PC has only been around for less than 15 years, the current level of processor and disk technology for less than 10 years, and high-speed fax/modems and Windows type interfaces, five years. The growth in computer technology is accelerating and prices are dropping.

Price/Performance

The cost of computing power drops about 30% per year, chips are doubling in performance power roughly every 18 months. In the last ten years, the cost per MIPS has dropped by 90% and shows no sign of slowing down.

Transactions Per Second

Another way of examining the cost of computing is to look at the cost of processing transactions in a given environment. This is a "real-world" way of judging the performance as it relates directly to business issues.

The Golden Rule

Given the dramatic fall in transaction processing costs, you should be able to pay for the new open system with just one year's maintenance costs for the old system. I have found this rule to be true for almost any proprietary system that is five years old or older.

Why Rightsize to Open Systems

Computers are faster and cheaper, but why are Open Systems the right way to achieve these savings? Why not purchase a less expensive mainframe or mini-computer without changing the application and environment?

With Open Systems everything becomes cheaper. The hardware, software, people, operating costs, etc. There is a much wider choice of vendors in a competitive market place, and a wider choice of product types. The mainframe market typically restricts the user to the products available from the machine supplier.

By Rightsizing, you can achieve much more with a lot less money.

Briefly, the savings can be divided into two areas: 1) hard dollar savings, and 2) some less tangible areas referred to as soft dollar savings.

Hard Dollar Savings

Examples of where hard cash savings will be made by rightsizing: virtually every aspect of system operation, energy costs, utility bills, and the cost of office space will be cheaper. A DEC VAX user recently moved an application to a newer, Open Systems platform. As a result, there was savings to office space accommodating two more employees. The new employees will be revenue-generating, replacing what was costing money.

Soft Dollar Savings

Faster processing speeds means you get your information quicker, people are able to achieve more, and they feel better about themselves.

Your various systems will be much easier to integrate, and information sharing becomes easier, reducing duplication of effort and frustration.

Your whole company becomes more flexible, leaner and fitter, and all of these benefits create much more confidence in the ability of the Information Systems departments to take on new projects.

Legacy COBOL Alternatives

We have examined the case for rightsizing. What are the options if we currently have an old legacy COBOL application running on a non-open platform.

Do nothing	–	Not a sensible option, but there some people doing just that.
RE-ENGINEER	–	Start again and restructure the business.
REPLACE	–	Buy an off-the-shelf package.
REWRITE	–	Be fashionable and go for the 4GL, 'C' etc.
MIGRATE	–	Rehost or relocate—just move it!

Application Alternatives

Re-engineering. Re-engineering means restructuring the entire business and all the systems to fit a completely new business model.

- expensive
- time consuming
- requires retraining of all staff
- requires changes to the business plan

Replace with Package Software. Off-the-shelf packages seldom meet business needs exactly. Consequently, they usually require a high level of customizing or the business practices need modifying to fit the package requirements. Users especially will need extensive retraining to use the new software.

Rewriting. Using 4GL's or 'C' means retraining everyone in new languages. No savings in either hardware or software costs can be made until the new application is up and running. This means paying the old maintenance for the duration of the project.

Application Alternatives, cont.

Relocating. The existing application is moved with minimum changes to the new platform by using the portability of ACUCOBOL-85. This is the easiest, quickest, and cheapest way of getting to Open Systems. It means savings, but does not preclude any of the other options from future considerations.

Agenda

How is a typical proprietary COBOL application simply moved to an Open Systems platform?

Alternative Migration Strategies

There are two basic ways of achieving migration:

Emulation Strategy

Emulation (by using a proprietary runtime on the new system to execute our existing code) only achieves part of the solution. We get to run our existing application on a cheaper machine, but get none of the future benefits of true open systems, and we are still locked into the emulator supplier. It is no coincidence that most emulators are typically supplied by computer manufacturers to lock-in their customers.

Pure COBOL Strategy

The solution is to use a pure COBOL environment using ACUCOBOL-85—a modern, portable, ANSI-85 compiler and runtime system. You can choose whatever platform you wish: UNIX, DOS, WINDOWS, etc., and change between them whenever you want to. No longer are you tied to a particular system supplier.

The rationale behind converting JCL and system calls to COBOL is that they also become portable for the future. For example, writing UNIX shell scripts means they will have to be rewritten or replaced in the future if you decide to move to Windows NT or OS/2.

Migration Process

Here are some simple rules to keep in mind:

- Use Migration Specialists who have proven tools and methods available.
- Start with simple applications, prove the technology, and gain some experience of using the new systems.

- There is no need to make major changes to the application—just move it to the new platform.

Migration Specialists

There are many Migration Specialist companies with extensive experience in moving Legacy COBOL Applications. Choosing the right one for you is the key to a successful migration project. We are looking for a mix of experience in many areas:

- in the enabling technology
- in the portable compilers and runtimes
- in knowing the differences between your old platform and the new one, and how to overcome the problems
- in a proven set of migration tools to automate as much of the process as possible
- in the new products available to enhance the application once converted, including future options
- in Project Planning and Management
- in help with training and support during and after the move
- in coordinating all of these areas

Enabling Technology

ACUCOBOL-85 is a completely portable language system. This is achieved by using a single-pass compiler to translate the source code into a pseudo-code. This pseudo-code is then executed by the runtime system. The runtime is specifically tailored to the platform it's running on, and continues as a native application.

ACUCOBOL-85 Portability

Once a program is migrated to ACUCOBOL-85, it will never need modification or even recompiling to move it to a new platform. We currently support over 600 different platforms. The new platform needs only the appropriate runtime to execute the compile pseudo-code. This is very important not only to software developers selling their systems to users with different platforms, but it also frees end-users from being tied to a particular supplier. This is the essence of Open Systems.

Where Do You Want to Migrate

There are now many choices of new platforms open to you. Acucobol has runtimes for all of these types and we stay current—as soon as a new platform or operating environment is introduced, we port to it. We are ready now for Windows 95.

Which File System

You also have a wide choice of file system: Acucobol has the ability to converse with several and they can be mixed within an application so it is possible to migrate to a RDBMS slowly if you wish. We recommend this approach unless there is a critical business issue demanding the immediate use of a database.

Which User Interface

Acucobol supports several of these environments and we have built-in tools for mouse handling, pull-down menus etc.

Acucobol provides the enabling technology for Open Systems. How do we get our old Legacy COBOL to ACUCOBOL-85?

Migration Tools

There are many tools used to perform migrations. They vary according to the old and new platforms, and between individual Migration Companies.

The Filtration Method

One of the most important tools any Migration Company can offer is a Code Analyzer. This program or group of programs parse the existing application source code, and identify non-standard or proprietary items that require conversion. Many of these Code Analyzers go much further, providing reports of redundant code, duplications; data dictionaries and cross-reference listings. The key is to know your existing application. Many legacy applications were written by programmers no longer employed by the same company, and their knowledge has been lost. Documentation is also frequently inadequate—the Code Analyzer can go a long way to replacing this vital information.

The main conversion filter is a master program which controls the COBOL Source Code conversion; it does not change from project to project and therefore is a known and proven tool. Each item identified by the Code Analyzer is the subject of a sub-filter program. In this way, each conversion task is kept small and manageable, and the work on these sub-filters can be carried out in parallel.

Depending on the old platform, we may also need tools to carry out screen interpreting, JCL interpreting, and menu handlers.

What Gets Migrated

There are three main areas that get migrated.

System Services	–	Screens
	–	JCL
	–	Report Sections
Data Storage	–	Files etc.
COBOL	–	The program source

System Services

System Services includes all those proprietary items that the new platform does not support or supports in a different way, the most common one being screen handling. Screen functions can be handled by coding COBOL routines to provide the same functionality as the original system, and calling them from the application in place of the original code. These routines can also be created by modern screen painters which have facilities to "import" older forms-type screens and create new screens and portable COBOL code.

Batch or Job Control is not commonly used on native UNIX, DOS or WINDOWS applications, as these systems are much more event driven, for example, by user requests from menus, etc. However, we do need to provide an alternative because many older systems make extensive use of batch processing. Typically, we would convert JCL into COBOL routines to keep them portable.

Data Storage

There are many proprietary file systems and they all require some form of conversion. The important point is that we must maintain LOGICAL RECORD INTEGRITY—the new file system must look the same to the application.

For example, assuming we were converting an IBM DB2 data base to a relational database such as ORACLE; we do not alter the design of the database; all our efforts are concerned with the differences between the syntax requirements of the two database engines. Our conversion sub-filters would translate the DB2 commands into SQL commands for subsequent processing by the Oracle Pro-COBOL pre-compiler.

Conversion of ISAM files, Sequential and Relative files is straightforward, provided our target platform and software environment have the necessary file system support; this is rarely a problem. Physically moving the data from one platform to another is time-consuming and requires careful planning. The old

platform will usually have ways of downloading data from databases and ISAM files to a common format (such as ASCII flat files). The new file system and/or the Migration Company will have tools for uploading such data to the new platform.

COBOL Syntax

The Code Analyzer identifies any syntax changes that need to be made, and subfilters are written to perform these changes.

The COBOL source code is passed through a series of sub-filters each designed to carry out a specific task modifying the original code as defined by the source list. This process is controlled by a main filter program which does not change from project to project, thus we have a proven base from which to work.

The cycle of conversion and validation is repeated until the tool is proven. Most migrations are not performed—they are written.

The Future System

Migration is an evolutionary method of getting to Open Systems, not a complete revolution. We can continue this process to evolve our application both visually and functionally with minimal code changes.

Evolve Visually

We can add features like pop-up windowing, color, and GUI's using the tools within the ACUCOBOL-85 language.

Evolve Functionally

We can add functionality to the application by adding client server with AcuServer, adding Hot-key programs for help programs, and we can redefine keyboards if we wish, without touching our source code.

Evolve Data Storage

It is possible to support multiple file systems. We can add a Relational Database either by using our seamless interface Acu4GL, or by using embedded SQL in conjunction with the pre-compiler from the database vendor.

Acucobol Benefits

By using Acucobol-85 as both the enabling technology to get to Open Systems, and as the development environment for the future, we achieve our objectives in a shorter timescale and at a fixed cost. We provide experienced project managers and we have a proven success record.

Implementation and Personnel Issues

- Planning
- Culture Shock
- Existing Resources
- Training
- New Resources

Planning

It is essential to have a clear plan of action for a migration project.

Everyone from the CEO on down needs to feel comfortable with the choices being made. It should not look like a forced move. Involve all levels of users: systems and programming, operations, and administration—they should have some input during the planning stage.

Pilot Projects

The choice of pilot project can really help if everyone involved sees the benefits of the move, and can feel confident about the outcome. Then the whole project stands a much greater chance of success. Pick a project that is neither small nor mission-critical. The purpose of a pilot project is to prove the technology and demonstrate the savings.

Project Milestones

Choose meaningful milestones for the project, especially during the negotiation with your Migration Company. The deliverables must be clearly defined, and responsibilities for those deliverables agreed upon. Make sure there is an agreed process for reassessing the project if the unexpected happens—Murphy says it will, plan for it. Make sure you define the end of the project. How to know when it is finished, what process needs to be performed to identify this, who signs off on what, etc.

Stick to the Plan

Having a great plan is only half the battle—it is equally important to stick to it. One of the reasons for choosing migration is that it is a short-term project, therefore it should be relatively easy to stay focused on the objectives. Above all, we must resist the temptation to experiment with any new developments that come along during the transition—there will be plenty of time for that when the migration is finished.

Culture Shock

Moving away from the "warm fuzzy" environment of the typical mainframe or proprietary mini-computer is a major culture change for most companies, and it requires careful planning to avoid alienating any part of the company. This is not just a cheaper, faster way of doing things, Open Systems have completely different ways of distributing information.

The organizational structure of IS will almost certainly change, pre-planned careers will need reassessing, and there may be layoffs. Operation duties will change dramatically, and there won't be a need for a centralized computer anymore.

Open Systems will make heavier demands on IS knowledge and skill sets. Individuals will need more knowledge about technology than in a mainframe environment. Mainframes tend to isolate people into niche activities, such as Shift Leaders, Tape Librarians, Console Operators, etc. Open Systems, because they distribute the technology around, involve the individuals in more "systems" activities.

Existing Resources

One of your most valuable assets during a migration is the knowledge and skills inherent in the existing staff. Maintaining and using their skills was one of the reasons we chose migration in the first place.

You will get some very mixed reactions from existing systems and programming people in particular, varying from, "Thanks for not throwing me on the scrap-heap" to "Oh No—not another three years of COBOL." Choosing jobs for these people both during the migration and afterwards is a real balancing act. However, there will be people who are so hide-bound by their old working methods that they will not be able to make the transition. Let them down lightly, but don't waste too much time on them. Typically, we find in most organizations that around 25% of the people will be totally gung-ho for the new system, about half will adopt a wait-and-see attitude, but regrettably some cannot or will not ever get the Open Systems message.

You should allow people to express their preferences for the future and accommodate them as far as possible, but remember, we are trying to make the company leaner and fitter—personal ambitions come second.

Be especially careful about alienating the people left to maintain the old system during the transition. Their contribution is vital, so make sure they have a career path after migration. Involve them in the training plan.

Training

You will need to develop an extensive training program for the existing staff. They will become the in-house experts and will be needed to guide future developments. Don't forget the end-users in this, if nothing else they may be getting new terminals, and they need to feel comfortable using them.

Initially, concentrate on platform specific training so that the new system is productive as soon as possible. Then move to training on the new compilers, runtimes, database and networking products. Finally, and probably when the migration is complete, train on the new development tools such as GUI, RDBMSs, Client/Server systems, etc.

Make sure everyone is involved in the design of the training program. This is their career development, as well as the future of the company that is being planned. However, do not allow personal ambitions to interfere with the business plan. The objective here is to make the transition to the new platform as quickly as possible; side-tracking and experimenting will detract from that.

New Resources

There are some areas where you are going to need experts—either just for the transition or for the future as well.

1) Project management needs to be clearly defined and the Project Manager needs to have extensive experience in handling Open Systems, multi-vendor projects, and coordinating internal and external resources.

2) Another is the Data Base Administrator, especially if you are going straight to an RDBMS such Informix or Oracle. You must have solid experience of using both the old and new databases to make a success of the move. There will be issues about how data is stored or moved on the different platforms.

3) You will need a good UNIX systems administrator for the new platform as UNIX can be a little unfriendly to strangers, not an area for a "trainee."

External Resources

The Migration Company should be able to supply some of these and any other new resources you need at least long enough for you to train your own from existing resources.

Your software vendors such as the compiler supplier or the RDBMS company (Informix, Oracle etc.) should be providing a great deal of hand-holding while you make the transition and afterward. Ask their advice in staffing up the project. Use their trainers and consultants.

Consider using independent contractors or consultants and learning from them as much as possible. This is also an area that needs careful handling if existing staff are not to become alienated. But, an important point to keep in mind is that, unlike on a new development project, using contractors does not dilute your knowledge pool. On a development project there is a risk that a contractor will come in, do a job and leave taking with them the knowledge of how the new system works. This is not true in migration - we're only going to do this once - we don't need to retain their knowledge about our systems as we already have it.

Quick-Change Artist

The primary reason for moving to Open Systems is to be able to adapt to change and to adopt technologies like Client/Server, GUI environments, and RDBMS's.

Case Study

Agenda

We have reviewed what, why, how and who. Finally, a successful case study is examined.

Here is a situation that faced a National Office Supply Retailer who had a large WANG VS 10000 system, and a substantial investment in heavily customized Wang VS COBOL applications.

Business Problem

The company opened their first store in 1989 and by mid-1992 they were opening 30 new stores a month. The rise in the demand for IS services was staggering and the WANG VS simply could not cope. Their overnight processing, which involved polling their store and downloading the day's information, was taking so long the management information was simply not available next day when it was needed. Also, with all the problems facing Wang they were losing

confidence in Wang's ability to support them; they decided they needed to move to Open Systems.

Choices

Their choices for redeveloping their old COBOL application were: ACUCOBOL-85 compiler and runtime system, an Open Systems hardware platform to run it on, and a migration company with experience.

Proof of Concept

In October 1992, a pilot project was chosen and carried out by what they called the "huge" test. The application software was converted through the migration tools only making sufficient changes to make it compile clean.

They moved one day's data to a borrowed Open System platform and ran their daily reports and analysis. Some programs crashed, some of the results were gibberish, but enough of the output was sensible and matched the Wang results exactly. The concept was proved and the performance indicators very favorable—at least 100% improvement.

Conversion

The next few months were spent refining and improving the conversion filters to take care of the anomalies noted during the pilot project. The UNIX system was installed in January 1993 and a simple communication link established to the Wang to allow the source code to be transferred back and forth.

The team consisted of in-house software developers, on-site specialists from the migration company, and management consultants to handle some administrative tasks such as impact analysis of some proposed system changes.

They repeated their "huge" test, this time downloading a full set of data on a Friday night, converting it to UNIX on Saturday, and running the analysis and reports on Sunday. It was a "huge" success—the program execution was nearly flawless, accuracy of results excellent, and performance still showed 100% improvement.

Switch-over

On June 18th all the data was converted and moved to the UNIX system, final tests run, and the Wang powered down.

Quote: "We held our breath, threw the switch and decommissioned the Wang the same day."

Quote: "Just one week after migration from the Wang we are up and running smoothly—users are delighted. I would never have predicted such a seamless transition could have taken place, but it has."

The Benefits

One year later, another 100 stores had been added, the IS load has doubled, but the UNIX system has lived up to its promise.

They are now making a move to Informix using Acu4GL—a seamless COBOL to SQL interface that requires no modification to the COBOL source.
Most importantly, portability has proven to be a major factor. Now they are considering a move to a different UNIX platform and this will be achieved by a simple change of runtime with no conversion.

Agenda

We've looked at the reasons and the methods for rightsizing. We've examined a particular migration methodology, how this can be implemented, and who would be involved. We've also examined a particular case study of a successful migration.

Presenter Profile

David Thompson was born and educated in the U.K. He started in the computer industry in 1967, and has worked on many different mainframes, minicomputers, and microsystems in a wide variety of industries. He immigrated to the U.S. in 1991, and now works as a Senior Technical Support Analyst for Acucobol, Inc. in San-Diego, specializing in migrations.

2. ACUCOBOL®-85 VERSION 2.3

In 1988 Acucobol, Inc. entered the COBOL tools marketplace with ACUCOBOL-85 and a deep commitment to highly portable, high performance COBOL tools. In the same year, Acucobol established itself as the leader in open systems COBOL. With the release of ACUCOBOL-85 Version 2.3, Acucobol again affirms its leadership position and enduring commitment to offering the world's most advanced open systems COBOL tools—tools for COBOL application development and delivery across diverse hardware environments.

ACUCOBOL-85 is an ANSI-85 compliant COBOL development system designed for portability, performance and reliability. The development system consists of the compiler, runtime, Terminal Manager, generic file system interface, Vision indexed file system, source debugger, and support tools and utilities.

The development system is complemented by AcuServer, network file access support for client/server environments; Acu4GL, a seamless interface to relational database management systems; and AcuView, an integrated business graphics package for COBOL.

ACUCOBOL-85 is supported by an expert development and technical support staff, and extensive documentation.

Portability

The ACUCOBOL-85 compiler is a single pass, pseudocode compiler that generates compact, machine independent object code. The object

code is interpreted at runtime by the ACUCOBOL-85 runtime module. The same object code can be run on any of more than 500 different platforms and seven different operating systems, including: most common variants of UNIX, MS-DOS, MS-DOS networks, MS-DOS with Windows 3.x, Windows NT, IBM OS/2, VAX/VMS, Open VMS, Data General AOS/VS and Alpha Micro AMOS.

At Acucobol, We Believe That Port-Ability is the Most Essential Element of Performance. If the Application Doesn't Run on the System, it Doesn't Perform on the System. There's Nothing to Measure; There's Nothing to Talk About. Applications that run on multiple platforms can be developed and maintained using a single set of source code. The source code is compiled only once. Portability doesn't require any recompilation. The same object file can be delivered for use on any platform supported by ACUCOBOL-85. For example, an application developed under MS-DOS can be run on a UNIX workstation without recompilation. This is accomplished first by the compiler, which generates machine independent object code, and second, by the runtime, which makes use of a host specific runtime configuration file to provide definitions for machine dependent values.

To simplify the task of converting existing applications to open systems, ACUCOBOL-85 is fully compatible with DG ICOBOL, VAX/COBOL and RM/COBOL. Compile time switches and special runtime support aids in the conversion of other COBOL dialects. For large or particularly complex conversions, Acucobol's Migration Services Division can provide any level of support needed to ensure the successful completion of your conversion project.

Performance

The ACUCOBOL-85 compiler compiles source code faster than any open systems COBOL compiler on the market. Average compilation speeds easily exceed 10,000 lines per minute. Error free compiles produce machine independent object files. ACUCOBOL-85 object files are ready for immediate execution. They do not require linking.

Superior compilation speed and the absence of a link step result in a very efficient code-compile-test cycle, which improves programmer productivity. Programmers spend more time engaged in productive work, and less time "waiting for the machine."

Portability Puts You in the Starting Gate. Product Performance Puts Money in the Bank. Application execution is managed by a compact and efficient ACUCOBOL-85 runtime system. SORT, MERGE, screen painting and file I/O support have been optimized for maximum performance. When system memory is very constrained, unused runtime utilities can be removed from the runtime.

Your application's runtime performance is not compromised by runtime interpretation. Applications that are highly computation intensive, such as systems software, language compilers, or modeling software, may not execute as fast as when compiled with some optimizing native code compilers. However, most COBOL applications are not computation intensive – most COBOL applications are I/O and memory intensive. For these applications, ACUCOBOL-85 frequently outperforms native code compilers because pseudocode is typically much smaller than native code (often as much as five times smaller). Smaller code uses less memory, which reduces memory swapping, improving performance and making more memory available for I/O and other processes.

Reliability

Reliability means that applications execute as expected on each and every platform supported. It has been our standard policy since 1988 that we will not ship a new release of ACUCOBOL-85 if there are *any* known bugs. Not one. The cornerstone of reliability is an efficient, bug free development system.

Offering a bug free development system is not just a claim we make, it is a reality experienced by our customers since we began selling ACUCOBOL-85. Today there are more than 2,000 developers and 200,000 end-users in over 70 countries using Acucobol products. Our reputation for reliability comes from them. You can find a sampling of their comments on page 6.

Fortifying ACUCOBOL-85 reliability is a dedicated technical support staff. We have assembled a world class team of COBOL experts who are ready to assist you whenever there is a need for technical support.

The Extended Development Environment

ACUCOBOL-85 is ANSI-85 COBOL, and a lot more. To help our customers build more powerful, flexible and user friendly applications, ACUCOBOL-85 includes many system extensions.

Compilation system extensions include:

- Graphical User Interface support for the development of graphical interfaces, including support for multiple windows, mouse actions, menu bars, pop-up and pull-down menus.
- Character-based windowing support. Allows the creation of windows, help screens, pop-up and pull-down menus, context sensitive messaging, window titles, and more.
- Machine independent, character-based display management. There is no need to write special code to support multiple display makes and models.
- An extended screen section that allows the specification of an entire screen format, layout and behavior which can be executed with a single DISPLAY or ACCEPT statement.

Runtime system extensions include:

- The Vision indexed file system. A fully portable, performance optimized, indexed file system that includes managed record locking.
- MS Windows and Windows NT runtimes. Instant native Windows applications including mouse support, multi-tasking, and Windows look and feel. These runtimes do not require that character-based application code be modified or recompiled. Most native GUI features are customizable and programmable.
- 64 bit architecture support. Your applications can run on the latest in super-scalar, highly pipe-lined, 64 bit processors. ACUCOBOL-85 is the first and only COBOL available for the DEC Alpha.
- Double byte character support for Asian character sets.

- COBOL and "C" library functions linked into the runtime. These routines include: functions to fetch error codes, copy and rename files, navigate the file system, handle the mouse and manage dynamic memory.

- An interactive, windowed debugger that supports three modes: source-level, symbolic and low-level.

Other extensions include:

- Utilities for examining object files.

- The vutil utility for maintenance of Vision indexed files. vutil provides support for creating, deleting, rebuilding, converting, examining and testing indexed files.

- The vio utility for copying and archiving data files. vio supports the grouping of files into archives, the extraction of files from archives, and the copying of files to other devices (typically tape and diskette).

Add-on options include:

- Network file access support with AcuServer™, Acucobol's file system for client/server.

- Acu4GL™ relational database management systems interfaces.

- Interfaces to a variety of file systems including Btrieve, MINISAM and C-ISAM.

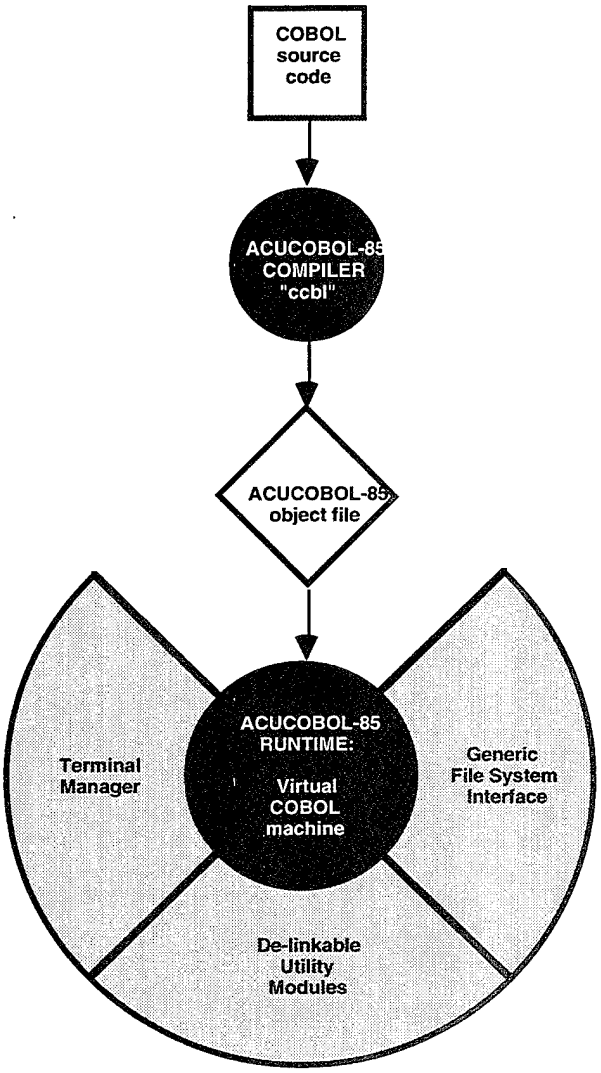
- AcuView®, an integrated business graphics and charting package.

In addition, a wide range of products is offered by Acucobol's Third Party Vendor Alliance, including: automated source code generators, screen builders, language sensitive editors, data file editors, and report writers.

System Overview

ACUCOBOL-85 is a complete development system for COBOL programmers building and deploying COBOL applications in open systems environments. ACUCOBOL-85 is designed to provide exceptional portability, performance, and reliability.

As the diagram below illustrates, at the core of the ACUCOBOL-85 system are the compiler and runtime. The runtime includes the interpreter, a machine independent Terminal Manager, the generic file system interface, an interactive source level debugger, optimized support utilities, and a host of library routines. On most platforms, ACUCOBOL-85 includes the Vision indexed file system.



Portability

Portability is primary to ACUCOBOL-85.

ACUCOBOL-85 is available on more than 500 platforms and seven different operating systems, including: UNIX (most variants), MS-DOS, MS-DOS networks, MS-DOS with Windows 3.x, Windows NT, IBM OS/2, Open VMS, VAX/VMS, Data

General AOS/VS and Alpha Micro AMOS. And Acucobol is committed to making ACUCOBOL-85 the first COBOL available on every new open systems platform as it enters the marketplace.

Programs compiled with ACUCOBOL-85 may be executed on any platform that Acucobol supports. Platform to platform, and operating system to operating system portability requires no recompilation. Multiple platforms can be supported with a single set of COBOL source code. Applications can be deployed across multiple platforms with a single set of object code.

With ACUCOBOL-85, your applications will never be restricted to a single platform, or stranded on an obsolete system.

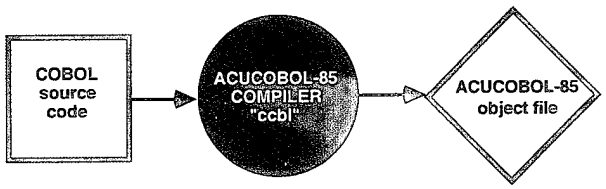
Performance

The ACUCOBOL-85 compiler consistently serves up the fastest compile times of any open system COBOL on the market. Average compilation speeds are well in excess of 10,000 lines per minute.

The ACUCOBOL-85 interpreter provides uncompromised runtime performance. Each runtime is optimized to provide exceptional performance on its host. Each runtime includes host specific optimizations to achieve the best performance for such activities as sorting, merging, screen I/O and file I/O.

The Compiler

The ACUCOBOL-85 compiler is a single pass compiler. COBOL source code is syntactically and semantically analyzed and tokenized. Errors detected during compilation are printed to the screen, or written to a file. Error free, tokenized code is written, in machine independent format, to an object file. The object file is ready for immediate execution using the ACUCOBOL-85 runtime module. No link step is required.



Portability Enabled by Pseudocode Technology

The ACUCOBOL-85 compiler is a pseudocode compiler, meaning that the object code produced is not targeted to a specific machine, but rather is targeted to a virtual machine. The virtual machine is a rigorously defined imaginary system that is capable of executing all of the actions required by ANSI-85 COBOL and

ACUCOBOL-85. The ACUCOBOL-85 runtime (runcbl) is an implementation of the virtual machine. The runtime's native code is ACUCOBOL-85 pseudocode. With pseudocode technology, Acucobol object code is made completely machine independent.

"Native" code compilers generate object code that will run on only one platform/operating system combination.

ACUCOBOL-85 masks most machine dependent attributes and values. For example, ACUCOBOL-85 provides machine independent data types. Designers and programmers need not be concerned with a particular machine's word size or byte order. Programmers needn't be concerned with the size or representation of pointer objects among machines. ACUCOBOL-85 data files and file handling are completely machine independent. And ACUCOBOL-85 supports 16, 32 and 64 bit architectures.

Many application developers are concerned that the overhead associated with interpreting pseudocode at runtime will impact program execution performance. ACUCOBOL-85 runtime performance is comparable to the best native code compilers. For most applications, program execution performance meets or exceeds the performance of competing systems. For a complete treatment of this important topic, please see the subsection titled Pseudocode interpretation in the Runtime section of this paper.

Using the Compiler

You invoke the ACUCOBOL-85 compiler on the command line by typing "ccbl". Source code errors encountered during compilation are output to the screen (or to a file, if one is requested). Each error message references the file name and line number where the error was detected. Compilation switches can direct the compiler to generate a source listing, symbol table listing, summary information and statistics. Other compilation switches support such options as COBOL source compatibility modes, source format options, reserve word options, data storage options, video options, debugging support, and a handful of other actions.

Source Compatibility

ACUCOBOL-85 is fully compatible with DG ICOBOL, VAX/COBOL and RM/COBOL. COBOL compatibility modes are selected at compile time via command line switches. ACUCOBOL-85 compiles in VAX/COBOL compatibility mode by default. Additional compilation switches aid in compiling other common COBOL dialects including: Micro Focus COBOL, MS COBOL, and CA Realia.

Compilation Speed

ACUCOBOL-85 is consistently the fastest ANSI-85 COBOL compiler available on any of the more than 500 platforms Acucobol supports. Why is the compiler so fast?

- Compilation is performed in a single pass (traversal) of the source code. Other compilers typically take two or more passes over the code.

- Generating pseudocode takes a lot less time than generating machine native code. The ACUCOBOL-85 compiler doesn't have the usual, time intensive, "back end" compilation phase that generates machine specific code.

- Machine specific optimizations are built into the ACUCOBOL-85 runtime and are not a part of program code generation. Developers don't wait, every time they compile the code, for the compiler to do machine specific optimizations.

- There is no link step. Eliminating the link step saves a significant amount of time over development systems that require compiled code be linked before execution.

Any application developer who has had to build a large application with a slow (or even average) compiler will confirm that compilation speed has a significant impact on programmer productivity and, consequently, on development schedules. The more efficient the code-compile-test cycle, the greater the programmer productivity.

Conditional Compilation

To support the highest degree of flexibility in writing and deploying applications for multiple platforms, ACUCOBOL-85 offers a combination of runtime and compile time support mechanisms. Most machine specific definitions are defined at runtime via entries in the runtime configuration file, cblconfig (for more information about the runtime configuration file, see the subsection titled Configuring portability in the Runtime section of this paper). For conditions that cannot be handled at runtime, ACUCOBOL-85 supports two powerful compile time mechanisms.

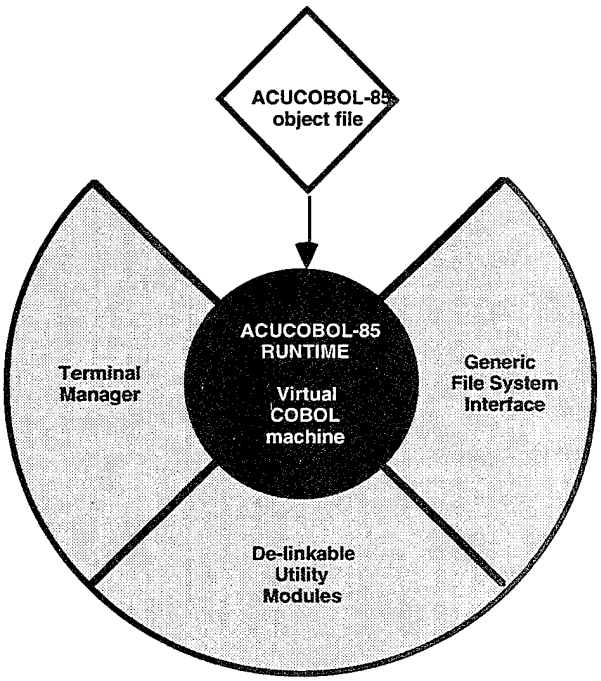
ANSI-85 COBOL includes, and ACUCOBOL-85 supports, the verbs COPY and REPLACE. COPY and REPLACE are directives to the compiler to insert or modify source code at compile time.

ACUCOBOL-85 also offers a conditional compilation mechanism in which specific lines of source code are marked with unique strings. Arguments given at compile time on the ccbl command line direct the compiler to include or exclude the marked lines from compilation. The conditional compilation mechanism allows multiple program versions to be maintained in a single set of source code. This mechanism can be used to accommodate multiple version requirements of all sorts.

Other Features

In the following the Runtime section, we've included detailed descriptions of a small selection of some of ACUCOBOL-85's most powerful features, including: Graphical User Interface support, the Screen Section, the use of runtime configuration variables, and keyboard configuration.

The Runtime



The ACUCOBOL-85 runtime provides all of the services required to execute ACUCOBOL-85 object code. Basic runtime system services include interpretation of pseudocode into machine code, dynamic linking of object modules, setup of code and data segments, memory management, and source level debugging facilities.

In addition to managing program execution, the runtime system provides a virtual operating system layer that effectively masks differences among host operating systems. The runtime provides such amenities as a consistent file naming convention, machine independent screen management, machine independent data file management, and a set of runtime library routines that are guaranteed to run the same on every platform we support.

A runtime configuration file, typically named cblconfig, is used to define host dependent values and to configure the runtime environment.

Pseudocode Interpretation

As the program executes, the runtime interpreter follows the thread of program execution and sequentially decodes each pseudocode instruction along the execution path. A typical pseudocode instruction may translate into one or more machine instructions.

The compiler attempts to minimize the cost of runtime interpretation by generating pseudocode which describes, as much as possible, large actions. For example, a "compare" instruction may translate into six machine instructions, while a "store" instruction may translate into a single machine instruction. Pseudocode instructions which interpret into many machine instructions are more efficient than those which interpret into few instructions.

Efficiency is a critical element of runtime performance. Some software producers are reluctant to use an interpretive system to deliver a commercial application. This is due, in part, to a belief that all interpretive systems are slow and inefficient. However, recent advances in processor power, improvements in software technologies, and our own innovative software engineering approaches have combined to give birth to a modern, efficient, competitive, and portable interpretive system: ACUCOBOL-85.

It is still true that applications that are extremely CPU intensive, such as systems software, language compilers, and modeling and simulation software, will not run as fast when interpreted as when compiled with a quality, optimizing, native code compiler (which, of course, produces non-portable object code). But how often is COBOL used to build such applications? It is common to use COBOL to build interactive business programs such as general accounting, inventory and manufacturing control applications. These applications tend to have a large number of interactive users accessing a common set of data. These applications do relatively little processing, and large amounts of I/O. These applications typically consume large amounts of memory and are often slowed by I/O subsystem performance and limited physical memory. Using an interpretive runtime system can actually improve the performance of many such COBOL applications. This is because pseudocode is usually a lot smaller than native code (averaging five times smaller). "On a system where memory is tight, replacing a native-code system with a pseudocode system can improve performance by freeing up memory, thus reducing memory paging. You are essentially trading away a plentiful resource – the CPU – for one that is scarce – memory." (from Portability Through Interpreted Systems, by Acucobol Chief Scientist, Drake Coker, in the August, 1993, issue of Unix Review. Request a reprint from your Acucobol sales representative.)

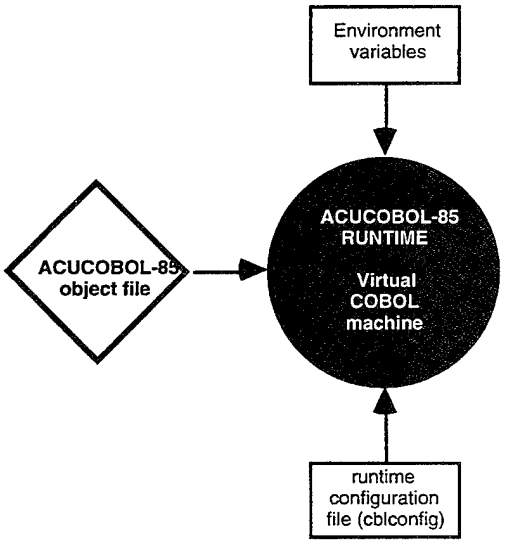
Configuring Portability

Machine dependent definitions, as well as many attributes of the runtime system, are controlled by configuration variables. These variables provide a great deal of power and flexibility in adapting an application and runtime to a particular system.

Configuration variables define such attributes as file name assignments, code and data file search paths, file status codes, terminal handling options, and file handling options.

Configuration variables are maintained in a standard text file that can be modified by the host system's text editor. Each entry in the runtime configuration file consists of a single line. All entries start with a keyword, followed by one or more spaces or tabs, and then one or more values. The configuration file, and all of its contents, are optional.

Runtime configuration values can also be stored in and retrieved from host operating system environment variables (on systems that support them). In addition, the COBOL program may define the values of variables during program execution using the SET ENVIRONMENT statement, and retrieve the values of some variables using the ACCEPT FROM ENVIRONMENT statement. For more about configuration variables, see the subsection titled Runtime configuration variables in the ACUCOBOL-85 Selected Features section of this paper.



Runtime Memory Management

When the runtime (runcbl) initiates, it requests memory from the operating system, sets up the program's code and data segments, loads the main program into memory and begins program execution. Subprograms are dynamically loaded into memory when first called. Once loaded, a subprogram remains in memory until it is canceled, at which point it is removed (this allows for full ANSI implementation of subprograms – variables retain their value when a subprogram is re-entered, files can be left open in a subprogram, and so forth). ACUCOBOL-85 includes extensions to the CANCEL verb and the PROGRAM-ID descriptor to give programmers more control over subprogram memory management.

Other tools are also available to aid memory management. Segmentation (overlays) can be used to reduce the size of a single program. The CHAIN verb and C$CHAIN library routine can be used to replace a running program with another program. Runtime memory buffers can be adjusted using options in the runtime configuration file.

In the MS-DOS environment, the runtime supports expanded memory (sometimes called LIM or EMS memory) when an EMS manager is installed. This allows very large programs to run under DOS. Acucobol also offers a version of the runtime that supports Extended Memory on MS-DOS machines. The Extended Memory version supports 32 bit addressing, providing addressing of up to four gigabytes of virtual memory.

On many UNIX machines, ACUCOBOL-85 supports shared memory. Shared memory allows multiple users to share the same copy of the program's object code in memory. This can dramatically reduce memory consumption and improve system performance by reducing the amount of memory paging that the system must do. For UNIX systems that support shared memory, ACUCOBOL-85 is shipped with a program named acumemd. acumemd is run as a UNIX daemon to support memory sharing.

Calling Other Languages

ACUCOBOL-85 supports calls to subroutines written in the C programming language. C subprograms are called by the COBOL program using the CALL verb. C subprograms may be called directly or indirectly. Direct calls pass parameters directly to the C function, simulating the behavior of a native code compiler. Indirect calls make use of an interface routine to pass parameters to the C routine (a sample interface is included with ACUCOBOL-85). With a C compiler, C routines may be linked into the ACUCOBOL-85 runtime.

In the MS-DOS environment, ACUCOBOL-85 also supports direct calls to assembly routines.

It is possible, using a C routine as a call interface, for your COBOL program to call any language that is callable by C.

Library Routines

ACUCOBOL-85 includes many standardized runtime library routines. Each routine is guaranteed to give the same results across all platforms. Library functions are called with the CALL verb. If required, any or all unused library routines may be removed (de-linked) from the runtime with a C compiler.

Version 2.3 library functions include support for such common activities as subprogram calls, file handling and operating system interactions, keyboard handling, screen handling, dynamic memory handling, menu handling, mouse handling, special support for MS Windows and Windows NT, and miscellaneous utility functions (text case conversion, program pause, and others).

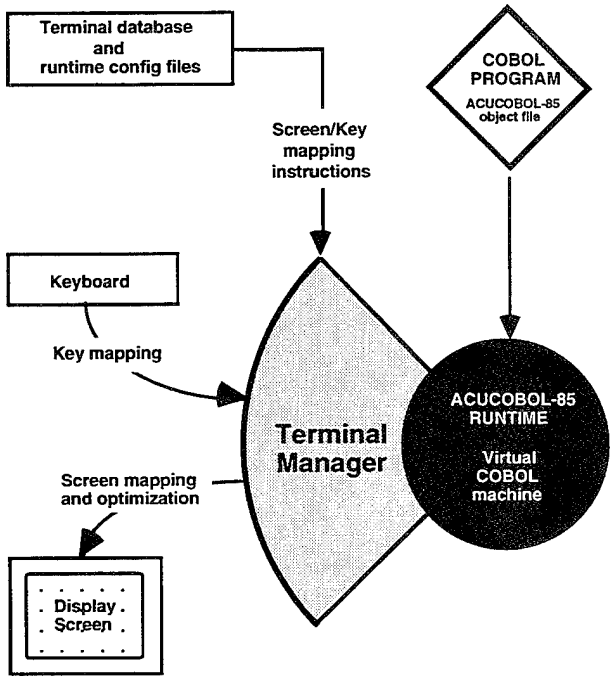
The Terminal Manager

The Terminal Manager module is a machine independent interface between the ACUCOBOL-85 program and the particular I/O devices present on any given system.

Utilizing a set of files and configuration variables, the Terminal Manager manages input from the keyboard and output to the screen. The Terminal Manager interprets the keys the user presses, translating each keystroke into a function. It also manages the translation of program display attributes from the ACUCOBOL-85 application program to the display screen.

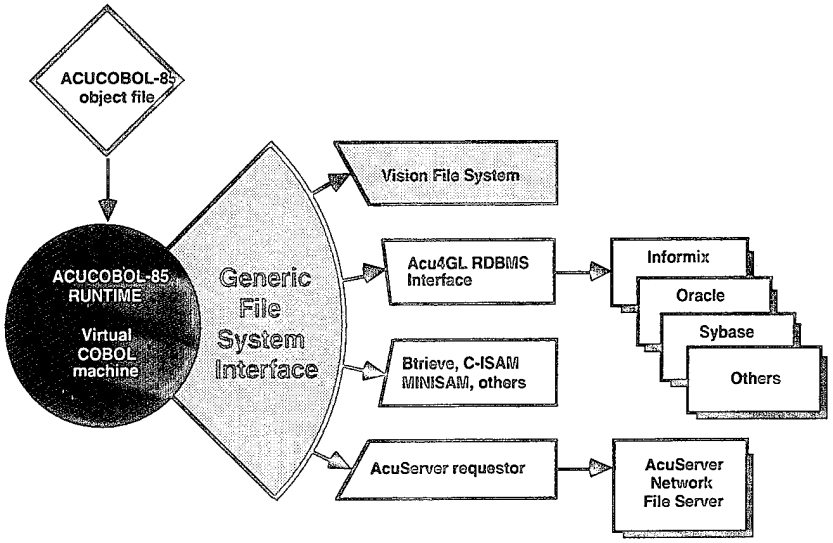
Through the Terminal Manager the programmer or user can specify the terminal type, designate a special action key, change the on-screen prompt character, add or change display colors, control data display formats and data entry formats, and sound error alarms.

The following diagram illustrates how the Terminal Manager relates to system hardware and software.



Generic File System Interface

ACUCOBOL-85 includes a generic architecture for connecting to file systems. This allows developers to choose from a variety of file systems and database management systems, as shown in the following diagram:



The generic file system interface gives application developers the ability to choose a file system (or a set of file systems) that best meets the requirements of the application, and the demands of the users. For instance, if the application demands a file system that is one hundred percent portable across DOS, UNIX and AOS/VS platforms, is performance optimized, and includes record locking support, use Acucobol's Vision file system. For excellent file system interoperability with other COBOL applications, consider C-ISAM. Btrieve is very well suited to DOS applications running on Novell networks. And MINISAM, on the AOS/VS platform, offers excellent performance without the need for data file conversion. Add Acu4GL if the application must interface to a relational database management system, such as Informix or Oracle . With Acu4GL you will get a seamless interface to the RDBMS without recoding your application or having to learn SQL.

Debugger

A development system is not a development system without a powerful debugger utility. ACUCOBOL-85 includes a full featured, easy to use, source-level debugger built into the runtime. The debugger runs in a separate window that overlays the running program without interfering with the application's normal screen I/O. The debugger supports three distinct debug modes: source-level, symbolic and low-level.

Debugging applications in source-level or symbolic modes requires that program source code be compiled with a special debug switch. You start a debugging session at runtime by specifying the "-d" debug flag on the runcbl command line.

The debugger supports a large set of capabilities including:

- Memory usage reporting, including: program memory, file memory, window memory, overhead memory (memory used by the runtime), and dynamic memory (allocated by the program).
- File tracing. File tracing reports file access activities as they occur.
- Paragraph tracing. Paragraph tracing reports the names of paragraphs and sections as they are entered.
- Shelling to the operating system.
- Script recording and replay. Script recording and replay is used to save debugging keyboard input and to replay script records.
- Program execution directives. Execution commands include: run, continue, go to cursor line, go until paragraph returns, go until program exits, skip to cursor line, step, p-step, and more.
- Source code navigation commands for fast navigation to a specific line or paragraph, for finding specified keywords, and more.
- Data object query and modification support for displaying the contents of variables, modifying the value of a variable, and monitoring a variable during execution.
- Breakpoint controls. Set and manage breakpoints, including the use of counters, on paragraphs, addresses, and lines containing verbs.

On systems that have mouse support, the mouse may be used to navigate the source and to select and direct debug actions including moving and scrolling through the source (in source-level mode), displaying variables, viewing procedures, setting temporary breakpoints, and more.

Debug Modes

Source-level: Source-level debugging permits the programmer to view and interact with the COBOL source code while the application is executing. Application source code is displayed in the debug window, and the programmer can interact directly with the source code to set breakpoints, inspect variables, step execute the program, and more.

Symbolic: Symbolic debugging allows the programmer to reference paragraphs and variables by their COBOL identifiers, but does not support viewing or interacting with the source code. Application object code compiled for symbolic debugging is much smaller than the same program compiled for source-level debugging. Some

software vendors deliver their application object code with symbolic debug support included. This gives the vendor a lot of power and flexibility when a problem requires debugging at the end users' site.

Low-level: Low-level debugging is supported for all object files, regardless of how the source code was compiled. To inspect an object in low-level debug mode, the programmer provides the absolute address of the data item in question. In practice, this means that it is essential that a compilation symbol table listing be generated when the program is compiled. The symbol table listing includes the absolute address of every program object. Note that the low-level debug mode includes support for the Trace Files command. Trace Files is a powerful tool for debugging data-dependent problems in complex applications.

ACUCOBOL-85 includes many capabilities that extend the power of ANSI-85 COBOL. Many of these capabilities have been introduced in the preceding Compiler and Runtime sections of this paper. Some of these capabilities include: COBOL source compatibility modes, conditional compilation support, library functions linked into the runtime, interfacing to other languages, machine independent display management, the generic file system interface, the Vision indexed file system, and the source debugger.

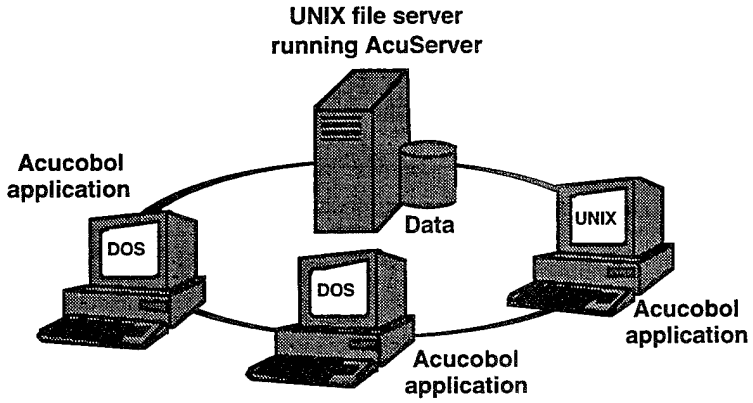
In this section we offer technically detailed descriptions of four select capabilities. Our intent is to include enough detail to allow for a complete understanding of each capability and its use in ACUCOBOL-85. In this section we detail ACUCOBOL-85 Graphical User Interface support, the Screen Section, runtime configuration variables, and keyboard configuration.

3. ACUSERVER

Acucobol, Inc., the leader in open systems COBOL, now offers AcuServer—network file system support for Acucobol applications running on UNIX and DOS TCP/IP networks.

AcuServer is a simple and economical way to give your COBOL applications remote file access support in client/server environments. With AcuServer, your applications gain:

- the ability to create and store data files on any UNIX system equipped with AcuServer
- full function remote access to all indexed, relative and sequential files from UNIX and DOS machines
- complete record locking support of indexed and relative files
- transparent access of remote and local files



No Recompiling

AcuServer does not require any recompilation of your existing programs. Your programs, including those compiled with earlier versions of ACUCOBOL-85, get instant remote file access capabilities when executed with a client enabled runtime and AcuServer.

How AcuServer Works

AcuServer provides remote file access support via a memory resident program (UNIX "daemon") named acuserve, running on a UNIX file server. COBOL applications, running on DOS or UNIX client systems, execute using an Acucobol client enabled runtime.

The runtime and acuserve work in tandem to fulfill remote file access requests. The runtime recognizes requests to remote files and bundles each request into a remote procedure call (RPC) to acuserve. Listening on the server, acuserve receives the request, manages its execution, and returns the result to the client requester.

A typical transaction might be:

1. An application running on a network machine attempts to READ a file.
2. The Acucobol runtime notices that the file to be read is located on a remote system and packages the request as an RPC to acuserve on the file server.
3. acuserve receives the RPC request from the client, executes the READ, and returns the record contents to the client, completing the RPC.

   Note that the READ access is accomplished with a single network transaction, thereby reducing network traffic and improving performance.

AcuServer places no limits on the number of clients or servers on the network (each file server requires a separate AcuServer software license). The number of clients a file server can support is determined by the processing power of the server and the demands of the clients.

Record Locking

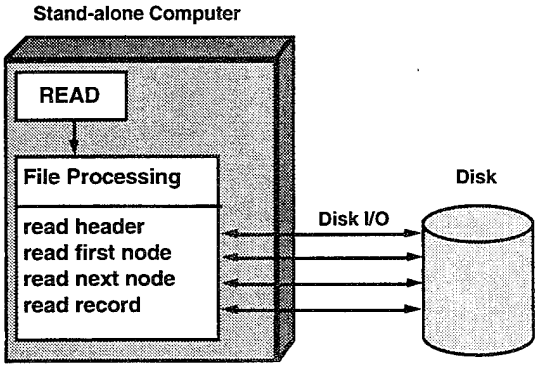
Record locking support is not compromised or restricted by AcuServer. Your indexed and relative files receive the same record locking support that is available in the stand-alone environment–any indexed or relative file opened in I-O mode can be record locked.

Performance

Network file systems often exact steep performance penalties on network file access. It is not unusual for network overhead to add fivefold, or more, to the time required to complete a common file action. AcuServer can't eliminate network overhead, but by using RPC, AcuServer can minimize it. To understand why AcuServer (using RPC) provides excellent remote file access performance, consider what happens when a record is retrieved from an indexed file.

Records in Acucobol's indexed files are stored in a balanced tree (B-Tree) structure. Reading a record requires a top to bottom descent of the tree. Each node traversed in the descent is examined for a match with the search value. The search terminates when either the desired record is found or the procedure has searched to the bottom branch of the tree.

Thus, several disk operations may be needed to complete a single READ statement. To perform the fetch in the least amount of time, the data file would need to be stored on the system's local disk, eliminating use of the network. Processing a READ on the local disk might look like this:



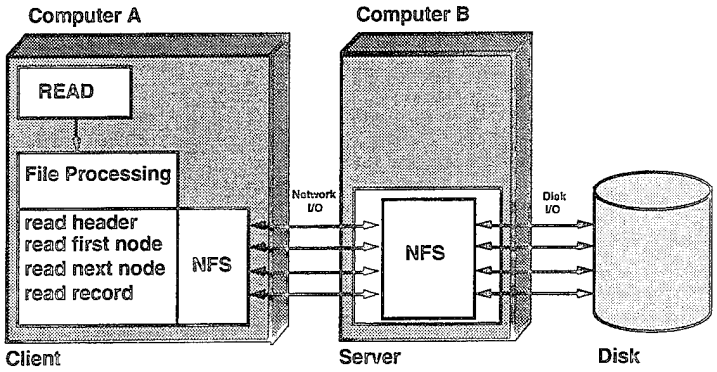
If speed is the only factor that matters, the dedicated system is hard to beat. But if data files need to be shared among many users and the users are spread out across many machines, some form of network file sharing system is necessary.

A popular and common network file service for TCP/IP networks is the Network File System (NFS). However, NFS has two significant drawbacks.

First, NFS record locking is not standard on all TCP/IP packages. Record locking support is missing from many NFS implementations. If you have a mixture of operating systems using NFS, you are likely to experience runtime errors if record locking does not exist in any one implementation, or if NFS record locking is implemented differently among the various operating systems.

Second, NFS places much of the processing burden on the client machine. There is a lot of network overhead because each disk access request is handled as a separate network transaction. As a result, many network packets must be sent and received to complete a single record access.

On networks using NFS, processing a READ request might look like this:

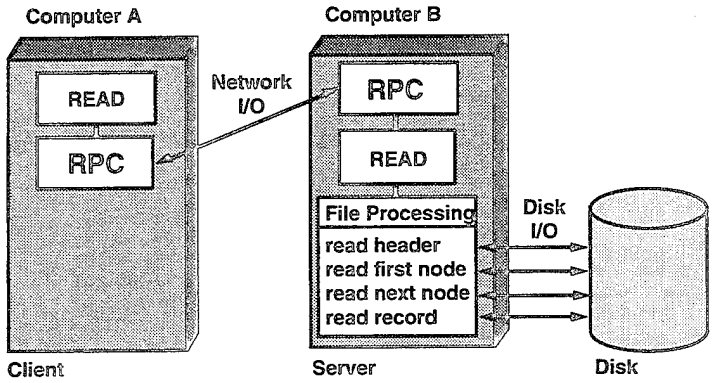


As the diagram illustrates, in addition to multiple disk operations you also experience multiple network operations. The negative impact on performance is significant.

AcuServer, using RPC, places the bulk of the file access processing on the file server.

The acuserve daemon receives a file access request from the client, handles the multiple file access actions needed to complete the request, and returns the result to the requester. There are no intermediate network transactions between the client and the file server.

Processing a READ using AcuServer might look like this:



The table below compares the costs of local, RPC and NFS file access. I/O timings were measured using Acubench. Acubench is an ACUCOBOL-85 COBOL program that measures the performance of sequential and random access file I/O on indexed files.

Acubench performance timings:			
Acubench	File System on:	Application running on:	Time (sec.)
Stand-alone	Sun	Sun	30.0
RPC	Sun	DEC Alpha	105.0
NFS	Sun	DEC Alpha	230.0

Security

To ensure uncompromised network and file security, AcuServer includes a transparent layer of remote file access protection. Using a site configured server access file, AcuServer validates client requesters for possession of access privileges before establishing a connection with the client. An optional password verification function allows you to assign a password to an authorization record. If the password option is on, the requester must supply a matching password before the client/server connection is established.

The AcuServer security system is designed to support any level of security requirements, from open and permissive, to highly restrictive, and most levels in between.

4. ACU4GL™ INTERFACES TO RELATIONAL DATABASE MANAGEMENT SYSTEMS

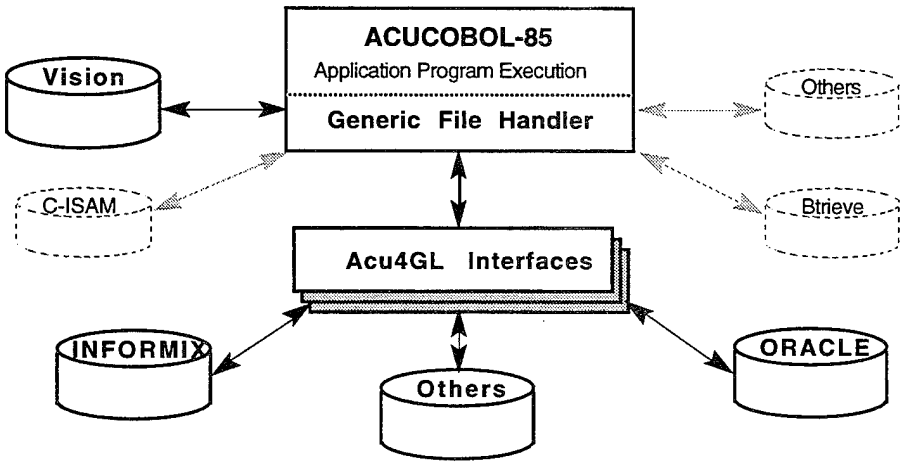
Acucobol, Inc., the leader in open systems COBOL, has developed the first product to interface seamlessly from COBOL to powerful Relational Database Management Systems (RDBMS) by dynamically generating Structured Query Language (SQL) queries.

With Acu4GL, users can benefit from RDBMS technology without learning SQL, rewriting their application, or spending a lot of time and money retraining. Acu4GL offers a simple path for adding the flexibility of RDBMSs to your existing robust COBOL applications.

Interchangeable File Systems

The standard file system supplied with ACUCOBOL®-85 is the Vision indexed file system from Acucobol. Vision supports variable-length records, data compression, and data encryption. Other indexed file systems can also be accommodated.

Acu4GL allows indexed file systems to be replaced by (or used in conjunction with) relational database management systems. This is possible because all ACUCOBOL-85 I/O passes through a generic file handler that can accommodate a wide variety of protocols by allowing different data storage formats to be "plugged in" as needed. The generic file handler determines what type of file organization is being addressed. Requests to access relational databases are passed to the appropriate Acu4GL interface module:



Accessing Databases

Acu4GL implements a direct, seamless interface between COBOL and RDBMSs.

Previously, accessing a relational database from a COBOL program involved writing SQL code and embedding that code in your program. You had to know SQL, and you had to write SQL statements appropriate for the specific database you wanted to access. Because your queries were tailored to suit one database management system, you had to recode your COBOL application if you wanted to access a different RDBMS, or an indexed file system, or even to migrate a file from one system to another.

As an alternative, some interface products now translate COBOL I/O statements into direct reads and writes on the database files, without going through the driver, or "engine," supplied by the database manufacturer. This means that the COBOL programmer must provide for enforcement of database rules that the engine already knows about and is designed to handle. Bypassing the database engine also means that new constraints or changes in the database structure will require reprogramming of the COBOL application.

Acu4GL implements a more suitable approach by dynamically generating industry standard SQL from COBOL I/O statements. As the Acucobol runtime module is executing your COBOL application, Acu4GL is running "behind the scenes" to match up the requirements and rules of both COBOL and the RDBMS to accomplish the task set by your application. This means that Acu4GL utilizes the full power designed into the database engine. The engine enforces database rules and constraints; any violations are returned to the COBOL program as I/O error conditions.

Acu4GL products provide a seamless, efficient interface between your program and the relational database. The interface is categorized as seamless because the communication between your COBOL program and the database is smooth, with no special query coding on your part, and no interruptions in the execution of your program. You need not change your COBOL code if you later want to access a different database or to access an alternate indexed file system.

The information exchanges between the database and the COBOL program are invisible to the end user. For example, if your program specifies a READ, a database query is automatically generated by the interface. Then the data that is read from the database is translated into a COBOL record. This exchange occurs in fractions of a second, and the application proceeds without interruption.

ACUCOBOL-85 communicates with relational databases via a special family of add-on interfaces called Acu4GL. Because relational databases manipulate fields, and COBOL programs manipulate records, some mapping is necessary to associate records with their fields. One function of the Acu4GL interface is to map COBOL records into database fields, and to map the database fields back into records. This is done by consulting data dictionaries generated by the ACUCOBOL-85 compiler. The next section shows how and when these data dictionaries are built.

Steps to Follow

The Acu4GL interface builds its own database queries dynamically whenever an input or output request is received. These are the steps that you take to compile your program and execute it using Acu4GL:

You compile your standard COBOL application with ACUCOBOL-85 Version 2.0 or later (the exact version depends on which RDBMS you want to access). When you compile, you specify via a compile-time option that you want the compiler to generate data dictionaries, in addition to an object code file.

An Acucobol data dictionary is created by the compiler for each indexed file in your program. These data dictionaries map COBOL records to the fields contained in the records.

In your configuration file, you may specify which RDBMSs you are using by setting the variable DEFAULT-HOST (for all files), or the variable <filename>-HOST (for individual files), or both. For example, you might say "DEFAULT-HOST VISION" and "EMPFILE-HOST ORACLE." This would direct EMPFILE input and output functions to the ORACLE RDBMS via Acu4GL, and direct I/O for all other files to the Vision system. These are runtime settings, allowing you to change hosts without recompilation.

Set 4. For different relational database systems, you may need to set up different procedures and configuration variables: login procedures, database names and locations, passwords and so forth. The appropriate Acu4GL interface manual and the RDBMS vendor's documentation will describe in detail the specific needs of each RDBMS.

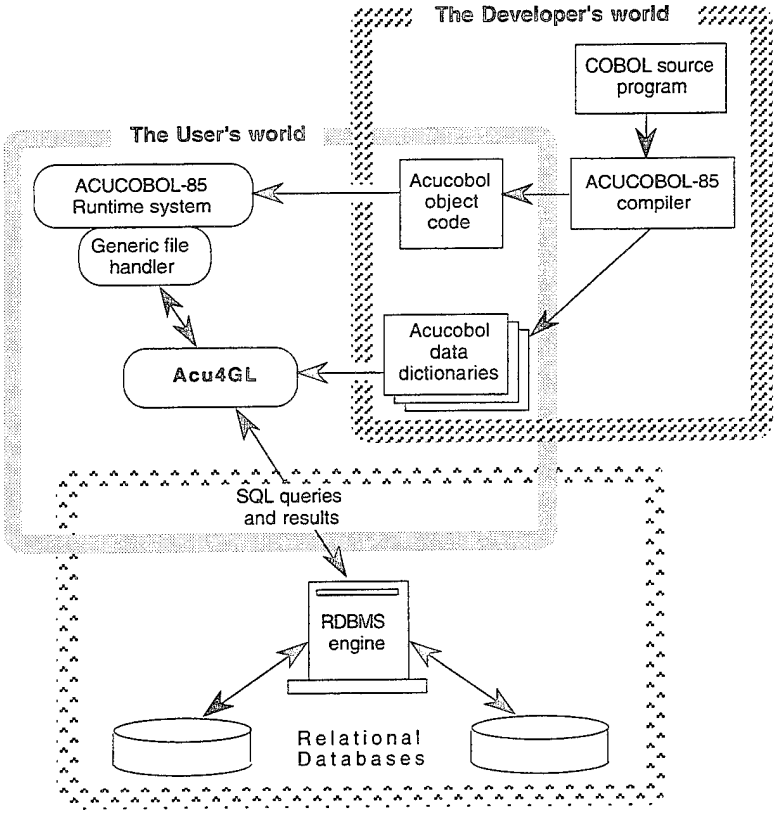
You use the Acucobol runtime system to execute your application. Whenever the runtime system encounters an input or output instruction (such as READ or WRITE) on a file that is directed to an RDBMS, it passes the request to the Acu4GL interface.

The interface automatically builds SQL instructions that your database management system can understand. As it builds these SQL instructions, it looks at the Acucobol data dictionary, which associates the COBOL records with their fields.

The database management system uses its own dictionary as a pointer into its own data files, performs the requested I/O operation, and passes the results back to the Acu4GL interface.

The interface translates the data fields into COBOL records or status codes, which are then passed back to the runtime system via the generic file handler.

This diagram shows how everything works together, from COBOL program to database:



Designed for Performance

All of the communication between the COBOL program and the database is automatic. All database queries and translations are performed behind the scenes, so that your end user experiences no interruption in program execution.

The Acu4GL seamless interface ensures that all changes to your database are immediately available to your COBOL program, and all data updates introduced by your COBOL program are immediately reflected in the database.

A high level of efficiency is maintained by a technique called "cursor caching." Prepared queries are saved in a parameterized format so that when the same function is to be performed on a different record (such as a "get next"), the task can be performed by executing the query with the new parameters, eliminating the need to regenerate the query. The concept is similar to calling a subroutine with arguments.

After your Acucobol data dictionaries have been generated, you can switch to a different RDBMS simply by adding a different interface module and changing your configuration variables. No recompiling is necessary, and (like Acucobol object files) the data dictionaries are fully portable.

Relational Integrity

Because Acu4GL accesses the database through its native "engine," the full relational integrity of the database is maintained. The COBOL program need not be concerned about enforcing relationships between keys and foreign keys on tables, constraints on field relationships and contents, and so forth. Database rules can be altered without necessitating a change to, or even a recompile of, the COBOL program that accesses the database through Acu4GL. If a database violation occurs, the engine detects it and Acu4GL returns an I/O error condition to the COBOL program. The program can, if desired, call a standard ACUCOBOL-85 library routine for an extended error description.

Ease of Migration

When you want to change the "location" of the data your COBOL program accesses, either from an indexed file to a database or from one database system to another, it is only necessary to acquire the Acu4GL Interface Product for that database. There is no need to recompile your program. You can also access more than one RDBMS system from your COBOL program, as long as you have the appropriate Acu4GL Interface Products. As another aid to orderly migration, Acu4GL also includes powerful utilities for moving data from existing files into tables, and for generating COBOL descriptions of existing tables.

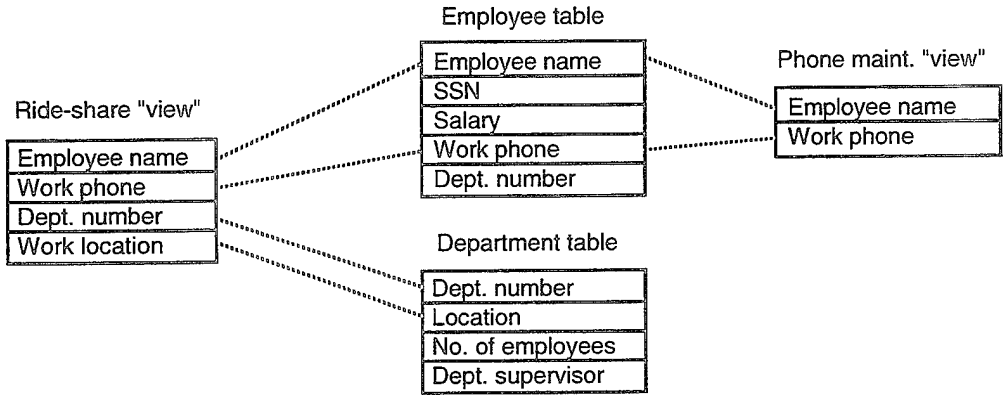
Controlling Data Access

Because Acu4GL is communicating directly with the database engine in standardized SQL, many of the more advanced features of the database system, such as "VIEWS," are accessible. The view feature can facilitate program conversion and ease future changes to databases. It can also provide control of user access to sensitive data.

A view (see diagram below) is a "logical record" that you define in your database system. It is a subset of one or more database tables. In the COBOL program, the same subset of data is described by a File Description. To the program, the "view" is just another file. When a COBOL read or write statement is executed based on that File Description, Acu4GL constructs the SQL needed to access the appropriate database fields.

An advantage of using views is that you can make any change you wish in the structure of your database, or even access your data from a different database on another RDBMS. Your COBOL program need not be changed as long as the view definition doesn't change.

It is also easy to preserve your control over user access to data when you move from one file system to another. For example, you might have a program that allows users to find out who lives and works close to them, for ride-sharing purposes. Your COBOL program searches an employee file and then displays pertinent data on those people who live close to one another, without revealing any unneeded personal information. To enforce the same access restrictions when your program is reading a relational database, you can define a database "ride-share view" that matches your COBOL File Description:



In some RDBMSs, there is a restriction that the fields within a view must be a subset of only one table in order for the program to "write" to the view, thereby updating the table. To illustrate, the "phone maintenance view" above accesses data solely from the Employee table, so a program using that view can both read and write that table. By contrast, the "ride-share view" accesses two tables, and can only be used to read from them. In summary, a view can be structured to control the COBOL program's access to database tables.

acucobol inc

Biography

DAVID THOMPSON
SENIOR MIGRATION SPECIALIST
Acucobol, Inc.

As Acucobol's Senior Migration Specialist, Mr. Thompson is responsible for assisting prospective Acucobol clients with proprietary COBOL migration issues. In addition, Mr. Thompson ensures project quality and supports migration management teams.

With more than 28 years in the computer industry, Mr. Thompson's professional career has included work in several industries, including banking, insurance, manufacturing, engineering, and commodities trading. Most recently, he was an independent consultant and was responsible for system design, implementation and operations. In this capacity he worked on a variety of mainframe, mini and micro platforms, including IBM, Honeywell, HP, NCR, Data General, and IBM PC & compatibles. In addition, he has a broad range of experience with software systems and programming languages, including UNIX, DOS, CICS, GCOS, COBOL, C, Informix, Assembler, and RPG.

A native of the United Kingdom, Mr. Thompson has been a featured speaker for industry conferences and seminars. In addition, he has addressed audiences at Acucobol-sponsored seminars in Hong Kong, Korea, Malaysia, and various cities in North America.

Acucobol, Inc. 7950 Silverton Avenue, Suite 201 San Diego, California 92126 (800) COBOL-85 (619) 689-7220 Fax: (619) 566-3071 E-mail: info@acucobol.com

REPOSITORY IMPLEMENTATION IN A LEGACY/REENGINEERING ENVIRONMENT

Amy King, Mark Koltz, Tanya Jones, Karen Stanford
Decision Systems Technologies, Inc.

1. OVERVIEW

While repositories are generally implemented with a long-term information management strategy in place, each aspect of a repository implementation should also take into consideration a specific organizational need. The information to be managed from the repository, the organizational need(s) the repository addresses, and the commitment to both the concept and execution of organizational information management, determine the success or failure of this approach. The implementation of a repository which both satisfies an immediate organizational need and which grows in depth and robustness over time, provides the organization with a body of information critical to the applications development and information systems management process.

2. THE OPEN REPOSITORY ENVIRONMENT

This section discusses repositories in general and the conceptual and physical environment in which a successful repository is implemented. It also briefly details the uses of repository technology to further objectives of shared and managed legacy applications information.

In order to successfully implement a repository, it is first necessary to understand the various complexities of repository technology and the organizational and technical issues they address. While repositories are generally implemented with a long-term information management strategy in place, each aspect of that implementation should also be based on addressing a specific organizational need. Once implemented, especially during the initial phases, the repository should not be evaluated based solely or strictly on the immediate need it addresses or on its immediate return on investment, but rather on its long term ability to help the organization achieve its strategic objectives.

Unlike other "tool of the moment" implementations, a repository and its associated tool-set are only half (or less) of the equation. The information to be managed from the repository, the organizational need(s) the repository addresses, and the commitment to both the concept and

execution of organizational information management scenario determine the success or failure of this particular approach. A repository provides an opportunity to define a process which enables a standard, repeatable, measurable, and distributable outcome. It additionally provides the vehicle through which to fully utilize the results of that process. While neither simple nor trivial, the implementation of a repository which both satisfies an immediate organizational need and which grows in depth and robustness over time, provides the organization with a body of information critical to the applications development and information systems management process.

Especially in the case of legacy systems, immediate and profound benefit can be gained through a repository implementation. A vast amount of organizational knowledge, as well as applications information, resides in legacy systems. This information relates both to the business of the organization as well as to the applications which support that organization in performing its day to day functions. Capture of the information currently encoded in legacy systems can provide a basis from which to maintain and enhance those systems other than through costly read-through-the-code analysis and maintenance. It further provides a platform through which to understand embedded business rules, and from which to move to a target platform or architecture in subsequent efforts, if that is part of the organization's overall I/S strategy.

In order to accurately capture legacy systems information without the requirement for extensive and error-prone manual input, a variety of tools can be utilized to extract, "clean up" and analyze legacy systems information, and to populate the captured information into the repository itself. This produces, even for legacy systems, a set of accurate, "clean", reliable information from which to maintain existing systems and develop new ones.

Unlike other applications development or information engineering tools, a repository is not a "stand alone" tool that accomplishes a single, specific objective. Rather, a repository implementation comprises an environment, both conceptual and physical, that is fundamental to its short and long term usability. This conceptual environment is based around three specific, broad concepts which, taken together, provide an organization with a fully functional, integrated information management capability. The concepts which underlie the successful implementation of a repository environment include:

- The repository itself - its internal toolset, its underlying database management system, its underlying meta model, its meta data population, and the "bridge" software through which information is loaded into or retrieved from the populated repository.

- The tools which provide input to the repository - application development tools, code generators and parsers, forward/reverse engineering tools.

- The policies, procedures, techniques and practices which support the repository implementation - data or process naming standards, repository update, access and security procedures, repository data base backup and other administration techniques and procedures, and repository administration practices.

The repository itself, along with its associated tools, data structures, and population, provide a place in which to capture, manage, and maintain organizational information, specifically applications design and implementation information. It additionally provides a vehicle through which to transition that information from its current state to some future state in a reliable, predictable, and thorough manner. In some cases, it is the meta model which ensures the manageability of certain types or "pieces" of information; in other cases, it is the data base structure, or the internal repository tool set. It might, in some cases, include the bridge software, which filters information before it is loaded into the repository. In all cases, however, the tools, concepts, and constructs which comprise the totality of "the repository" are critical to its implementation.

Similarly, the applications development tools which interface with or interact with the repository are crucial to the successful implementation of the environment. Applications and design information are critical pieces of information to the organization. New design information must be documented; application changes must be managed and the impact of change assessed; and systems must occasionally be redeveloped or retrofitted, often with automated assistance for at least some part of the transition. The concept which underlies this aspect of the repository environment is that an organization's need for specific tools changes over time, as does the number and quality of tools available in the marketplace. As a result, an organization requires the ability to select the "best of breed" tools for a given task or set of tasks, and to augment, or replace those tools as required without requiring a re-do of the applications information addressed or produced by those tools. When these tools communicate to and through the repository, the capability to select, change, and eliminate tools or tool sets is much more quickly and cleanly accomplished.

Finally, the organizational aspects of the repository implementation comprise a vital component of the repository environment. Because the concept of shared, managed information resources represents a fundamental change in the way most organizations develop applications systems, the administrative and procedural support for the repository environment ensures the reliability, accessibility, and integrity of the effort, as well as of the repository population itself. Policies must be put in place to establish who will have access to the repository, and who will be authorized to update its contents. Like any other data base, the repository must be backed up, tuned, and monitored for performance. Information being loaded into the repository must be valid

and error free, and procedures must be established to define how this will be accomplished. Only when the appropriate procedures and practices are in place can the organization be assured that its repository environment is valid, reliable, and secure.

As stated earlier, the repository tool itself is only half (or less) of the equation. A true, fully developed repository environment provides the capability to not only store and retrieve information, but also to actively manage and maintain the totality of the applications development environment. A repository implementation, while complex, provides comprehensive, reliable information to all levels of the information systems organization, and assures that the organization will always be fully in control of its information, its existing applications, and its transition and design efforts.

3. ROLES OF THE REPOSITORY IN A LEGACY/TRANSITION SCENARIO

As discussed in the previous section, a comprehensive repository implementation can provide an organization with an accurate, reliable set of information about its information systems - both legacy systems and those that are in some form of development - transitional, new, or under maintenance. Because of the flexibility inherent in repository structures, and because of the comprehensiveness of the information contained in a fully populated, robust repository, the repository can fulfill many roles within an I/S organization.

While the transitional, design, and new development aspects of repository usage are beyond the scope of this document, it is clear that only through the use of repository technology can the concepts of code reuse, data warehousing, model management, and other organizational information management objectives be fully realized.

Within the relatively bounded area of legacy systems, however, significant contributions from the repository are also possible. With the capture and central management of information about existing systems, and the relationships among that information, the accomplishment of a number of organizational objectives can be accomplished, and a significant facilitation of the information systems development process can occur. The areas in which a repository, populated with a comprehensive set of legacy systems information, can significantly contribute to the applications development and information systems management efforts center around the following areas:

- Impact Analysis. One of the areas in which a repository can immediately and profoundly benefit individual developers as well as I/S organizations as a whole, is in the area of impact analysis. Unlike code analyzers, home grown cross referencing tools, or other stop-gap measures, a fully populated repository can provide both very general and extremely detailed information for performing

impact analysis. The relationships established among repository entities provide critical "where used/how used" information, not just within a specific program, application, or piece of code, but across multiple systems, platforms, languages and tools. Every individual occurrence of a copybook usage, for example, can be quickly ascertained via the repository, providing "scope of change" as well as impact information to maintenance programmers and information systems planners.

- **Configuration Management/Change Management/Version Control.** As with the subject of impact analysis as discussed above, the inclusion in the repository of detailed information about legacy code (including information about data elements, literals, processes and procedures, copybooks, etc.) provides an extremely fine level of control over changes which occur within that code, both within and across systems, platforms, and tools. It additionally provides the capability to track changes made selectively to some (but not all) occurrences of a specific repository entity. Changes made to a specific data element, for example, can be identified, assessed for correctness as well as impact, and managed. If that change affects only certain occurrences of that data element but not others, those affected can be assigned a different version from those which are unaffected, providing a path into the changed elements, the unchanged elements, or all elements, depending on the information desired. The different versions of the element can be managed separately, or all occurrences of the data element can be managed without reference to version, depending on the application for that data element in a given situation.

- **Data Standardization.** Again, this is an area in which sweeping changes can be made, greatly improving the quality and reliability of both applications code and business data. The capture of data element information, especially if that capture is from an automated source (which provides more accurate and reliable information than manual data entry methods) can immediately highlight inconsistencies, errors, and other flaws in an organization's data element population. While many organizations already have some form of data dictionary or data naming standard in place, the repository highlights ALL occurrences of ALL data elements, along with the context (where used/how used) in which they exist. Again, this provides cross-system, cross-platform information, enabling an organization to standardize and improve data naming and data quality organization-wide.

- **Documentation.** A not so widely recognized but nonetheless important feature of a repository implementation is the ability of the repository to provide online, accurate, "living documentation". Again, especially in the situation where the repository is automatically populated and updated, information about legacy systems objects (copybooks, programs, calls/called by sequences, etc.) is as

accurate as the most recent parsing of the source code. The manual movement and synthesis of applications information from an electronic medium into hard-copy documentation is inherently error-prone. It is also inherently "obsolescence-prone", in that it generally happens that only after all code changes have been made does any documentation get updated (if then...). As a result, documentation tends to be at least "one generation behind" the realities of what is actually contained in the code itself. This would appear to be of little consequence, except that, other than the code itself, documentation (if it exists) is the primary reference used to perform impact analysis and code maintenance. The advantage of current, automated repository information which replaces some or all hard-copy documentation, especially for emergency, complicated or critical maintenance efforts would seem significant.

4. REPOSITORY IMPLEMENTATION EXAMPLE

Recent and successful repository implementation projects have been performed with the Internal Revenue Service (IRS). Decision Systems Technologies, Inc. (DSTI) is currently analyzing IRS legacy systems in support of the IRS Tax Systems Modernization (TSM) program. As part of the project, DSTI delivered a fully functional repository and meta model using MSP's Open Workgroup Repository, Oracle 7.1, UNIX, MicroFocus Revolve, Trinzic Forest & Tress, and a client/server application platform. The reengineering workbench required the analysis of over 1.25 million lines of COBOL code and complete cross referencing of all underlying aspects of system and technical documentation. Attributes and data models of the application software were inventoried in the repository to document and help manage software transition. The automated and seamless transition from source code to the Open Workgroup Repository delivered "Living Documentation" that was complete, accurate, comprehensive, and synchronized with actual production application source code.

In the next phase of the project, DSTI will conduct a comprehensive assessment of the remaining components of the system (approximately 20 million lines of code). DSTI will develop strategic redevelopment recommendations to leverage existing applications, reuse systems and program components, improve the integrity and completeness of new systems, streamline the application development process, and to facilitate application integration efforts. We will also develop a re-engineering feasibility report based on the technical and functional assessment of the system.

5. CONCLUSION

While it is beyond the scope of this discussion to list and describe all possible uses for the information contained in a repository, it is clear that even a less-than-complete or legacy-only implementation of a repository can provide significant benefit within an I/S organization. The critical differences between an open repository approach and other less comprehensive efforts, is the ability of the repository to accept automated input, and the ability to establish and utilize the relationships among repository information across platforms, languages, tools, and organizations.

BIOGRAPHY

AMY KING
Vice President and General Manager
Decision Systems Technologies, Inc.
6301 Ivy Lane, Suite #600
Greenbelt, Maryland 20770
Phone: (301) 441-3377, x 6347
Fax #: (301) 441-4571

Ms. King has over twelve years experience in the management, design, development, testing, and integration of large scale software systems, with extensive experience in team building and management of software analysis and engineering groups, to include direct management responsibility for major software requirements, scheduling of activities, and tasking of personnel. Her experience includes the development of systems using a variety of hardware in government and military logistics applications. This experience covers all phases of systems development and spans systems programming; executive information and database management system development; development of command and control software using rapid prototyping technology; configuration management, IV&V, quality assurance, and development of documentation using standards.

Ms. King currently has management, development and quality assurance responsibilities for DSTI's Integrated Systems Group with a staff of 70 at several different locations, providing systems engineering, program management support and system development activities with special expertise in rapid application development, information engineering and redevelopment, and IV&V.

Automatic Data Extraction from Free Text Messages

Chris McNeilly, Louise Osterholtz and Richard Lee
Sterling Software, ITD

and

Dr. John Hermansen
Language Analysis Systems, Inc.

August 30, 1995

Abstract

The Counter Drug Intelligence System (CDIS) is an integrated information system supported by text files and a database that requires accurate, near-real-time data to support its user community. The Automated Templating System (ATS) was developed to expedite the process of extracting useful information from free-text electronic message traffic and updating the CDIS database via sets of database templates. Using Lockheed-Martin's NLToolset Lisp code as a base, we have developed a system that will fill in database records automatically for this complex, event-oriented database structure.

The ATS currently handles four distinct counter-narcotic message types. An interface has been provided for human validation of the updated records. In order to accommodate the increasing demand for templating additional message sources, an Automated Templating Assistant is being built to assist analysts in the creation of correct and complete database updates from unrestricted message types.

Our test of the system's accuracy compared ATS against seven experienced analysts on a set of 25 messages, and found the system to be both much more thorough and much more consistent in its creation of data than any of the human templaters. Of course, it is also much faster and operates around the clock producing upwards of 40,000 useful database inserts during one month of testing. By the government's estimate, this volume of database updating represents the equivalent of 13 analysts manually entering the data over the same time period.

The ATS was built for the DIA, originally under JNIDS sponsorship. It is currently fully operational, with enhancements and extensions in progress. The effort required 3.5 persons with advanced training in computational linguistics, and covered the period from 3/93 through 9/95. The success of the project has resulted in several extension options, including definition of word processing formats that would expedite templating and database updating of documents outside the normal message stream.

1. INTRODUCTION

The principal problem for intelligence analysts in the information age is simply stated: too much data and not enough time to analyze it all. Relational databases were created to organize and sort data, and elaborate retrieval tools were created to make the data accessible and useful. Data entry, if it was considered at all, was deemed a time consuming but trivial task. In fact, database systems have been developed but never utilized, simply because the database was never sufficiently populated. Early in the development cycle of the Counter Drug Intelligence System (CDIS), intelligence analysts realized that the amount of data they were receiving daily was more than they could handle and some automatic means were necessary to create database records from information in free-text messages. The Automated Templating System (ATS) addresses this need.

This paper will discuss the ATS and its role in the CDIS project. First we describe the CDIS system, including details on the database structure. Next we give details on ATS processing. Then, we compare the results of ATS and human analysts templating the same messages. Finally, future directions for the ATS system are discussed, including the Automated Templating Assistant and expanding ATS-technology to other tasks.

2. CDIS OVERVIEW

The Counter Drug Intelligence System (CDIS) is a detailed data exploitation system that provides analysts the capability to receive, translate, understand and route data from a number of sources. The primary data sources are electronic messages and scanned documents, all of which are free text. The information is stored in a Sybase relational database. The system was developed in a rapid prototyping environment, with deliveries and demonstrations to the end users approximately every two months. Enhancements and improvements were driven by the analysts themselves after each cycle. This section provides information on the CDIS system and its domain. A more complete description of the database structure is given to show the level of detail required in the ATS system.

System Domain

When describing their domain, counter drug analysts repeatedly speak in terms of events. Therefore, the CDIS database schema utilizes an event-oriented design, where people, places, and things are players in events. For example, an analyst might identify a purchase event and then identify a buyer, seller, and commodity that was purchased. In addition to people and places, analysts need to track organizations, facilities, aircraft, vessels, and materials. Various relationships between entities, such as ownership, and family relations or organizational affiliations, are also of interest to analysts.

System Components

In addition to ATS, the CDIS system contains many other components, ranging from data capture and management tools to data exploitation tools modules. A brief description of each component follows.

- **Automated Message Handling** software utilizes user-defined profiles to route message traffic to analysts and to ATS. Data sources include AUTODIN and Press message traffic, and soft- and hardcopy English and Spanish documents.
- **Optical Scanning/Archival Storage** system scans, recognizes and stores hardcopy documents at the rate of 10,000 pages per 24 hours. Storage capacity is nearly one-half terabyte on re-writeable optical jukebox disks, and Excalibur's EFS filerooms are used to manage the document collection.
- **Machine Translation** component, based on the SYSTRAN translation system, provides analysts with rough translations of Spanish documents with the click of a button.

- **Manual Templating** tool allows an analyst to browse the messages in his/her queue and then extract important information and update the database. The tool is oriented around the idea of an *event*, where the user first defines the *event* and then adds *entities* that fill particular *roles* for that event.

- **Document Retrieval** provides analysts with a graphical interface to invoke the EFS full-text search function against any of the filerooms of softcopy documents.

- **Database Retrieval** allows analysts to query the database based on entity information and/or event information, via a graphical user interface.

- **Link Analysis** using NETMAP accesses the CDIS database to give a graphical wagon-wheel representation of relationships between entities and events.

- **Predictive Analysis System**, developed by SRA Corp., uses narcotic operation scenarios and event data from the CDIS database to predict future events. This expert system relies heavily on the event-oriented schema.

- **Temporal Analysis System** (TAS) is used to analyze activities over time in order to establish patterns of behavior. TAS uses a timeline representation to show temporal relationships.

- **Mapping Applications Client/Server** (MACS), developed by Sterling Software, lets analysts perform operations such as zoom, pan, and feature visibility selection on a geospatial display. Geospatial analysis functions are also included.

Database

The CDIS database is a relational Sybase database with approximately 150 tables. For a typical message, ATS will update more than 40 tables. Conceptually, the structure can be thought of as representing entities, events and relationships between them. Table 1 lists the different types of entities and events that ATS must be able to extract.

Entity Types	Event Types
Account	Agricultural
Address	Arrest
Aircraft	Communication
Callsign	Construction
Coordinate	Contracted Service
Document	Financial Transfer
Event	Investment
Facility	Loan
Individual	Meeting
Material	Planning
Organization	Processing
Telephone	Purchase
Vehicle	Relocation
Vessel	Seizure
	Storage
	Transhipment

Table 1: CDIS Entity and Event Types

Extensive detail may be extracted for any entity. For example, for an individual, ATS can update physical information (hair color, height, weight, etc.), as well as educational background, military history, passport information, residence, occupation, etc.

Each event has specific mandatory and optional roles which entities fill. For example, a Transhipment event has Sender, Receiver, Agent, Carrier, Conveyance, Origin, Destination, and Cargo slots; the Cargo role must be filled by a material while the other roles are filled optionally. The types of entities which can fill a role are restricted; for example, only vehicles can fill the Conveyance role in a Transshipment event.

Additionally, over 120 entity-to-entity associations can be recorded. For example, an individual can be the Owner and/or Pilot of an aircraft.

3. ATS PROCESSING

There are four basic steps involved when ATS processes a message: **Pre-Processing, Pattern Matching, Reference Resolution and Discourse Processing,** and **Post Processing**. Each step is briefly described below, using the following text as a typical example of the messages which analysts and the ATS must process.

> ON 27 JANUARY 1995, PILOT XAVIER JOSE VELIZ-CRUZ, DOB/120749, CIT/US, WAS ABOUT TO DEPART DONALD SANGSTER INTERNATIONAL AIRPORT FOR MIAMI, FLORIDA ABOARD A/C YJ123P WHEN A SEARCH BY JAMAICAN CUSTOMS OFFICIALS REVEALED 12 POUNDS OF MARIJUANA CONCEALED IN HIS LUGGAGE. VELIZ-CRUZ IS THE REGISTERED OWNER OF YJ123P.

Figure 1 diagrams the relationships between entities and events mentioned in the example text. In this example, there are three events (depicted as boxes) that relate various entities (ovals) to each other through relationships (the labels on the lines). Also, relationships can be drawn from one entity to another, such as the *owner* relationship between individual *Veliz-Cruz* and aircraft *YJ123P*.

Pre-Processing

The first step in ATS data extraction is to identify messages to process. The message handling component of CDIS identifies messages of the types that ATS can process and routes them to the ATS message queue. ATS periodically checks its queue and begins processing each message in turn.

Pattern Matching

The bulk of ATS processing occurs in the pattern matching and reduction phase. The system makes numerous passes over the message, and at each pass attempts to group words and the results of previous passes together into meaningful chunks. For example, the first pass might try to recognize organizations, so the phrase "ABC Widgets Company" is identified and reduced to the single token "*org*"; later passes only see the simplified token. After all of the entities have been recognized and reduced, the pattern matcher then tries to match entire clauses and generate events and relationships. The first sentence of our example text will be reduced, after several passes, to:

> ON *DATE* *IND* WAS ABOUT TO DEPART *FAC* FOR *ADD* ABOARD *AIR* WHEN A SEARCH BY *ORG* REVEALED *MAT* CONCEALED IN HIS LUGGAGE.

With entities simplified to single symbols, event and association information is much easier to recognize. In this case, the **Transhipment** and **Seizure** event patterns and the **Pilot** association patterns are matched, which triggers or *activates* the appropriate event and association templates.

The different passes of the pattern matcher are both linguistically and pragmatically motivated. In general, the reductions attempt to adhere to phrasal boundaries such that all noun phrases are reduced prior to the clause-level

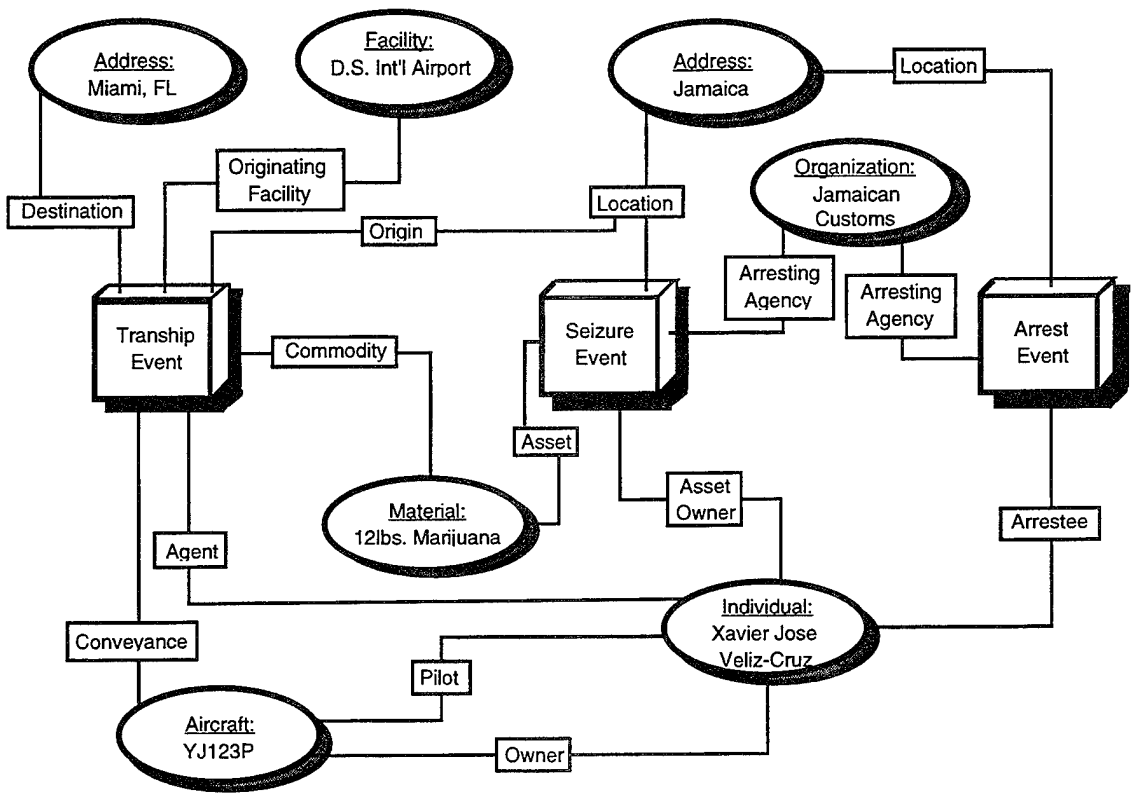


Figure 1: Relationships among entities and events

passes which uses verb sub-categorization information (the verb *sold* requires a subject and two objects, which fill the roles *seller, commodity, and buyer* respectively) to realize the relationships between entities and events.

Reference Resolution and Discourse Processing

ATS pattern matching operates on a sentence by sentence basis, so if two sentences mention a seizure, two filled event templates will be generated. After all pattern matching passes are completed, the discourse engine collects all of the templates activated by the pattern matcher and attempts to collapse and merge templates that represent the same entities, events, and relationships. To accomplish this, the tool uses *contexts* to help control merging. Contexts can be defined either tightly, where almost any new activated template will cause a context shift, or loosely, where only conflicting templates would cause a shift. Only events within the same context are candidates for merging; entity templates merge when no conflicting information is present.

Post Processing

The final step in ATS processing is the post process, which performs two main tasks.

- Pronominal reference resolution. Pronouns in messages usually refer to previously mentioned entities; the first phase of post processing is to determine which entity a specific pronoun points back to. For example, the pronoun *it* may refer to any one of a number of entity types (aircraft, vessel, vehicle, or material); the post

processor identifies the closest previous occurrence of an entity of that type. That entity is then inserted into the event or association template in place of the pronoun. Pronominal reference resolution, or anaphora resolution, can be a message-type specific task: at least one message type uses *he* or *she* to refer to aircraft and vessels.

- Data Base Update. The second phase of post processing converts the set of activated templates to Structured Query Language (SQL) statements which are passed to the CDIS database engine. Thus CDIS database records are created, and the extracted information is added to the database.

4. RESULTS

In order for analysts to have confidence in the database records which ATS creates, an analysis of ATS performance is critical. To effectively evaluate ATS performance, the system's results must be compared to the set of *correct* templates for a set of test messages. The problem lies in agreeing on what the correct templates are for a given message. Seven analysts were each asked to template 25 messages, and the resulting templates were to be merged to create a baseline for evaluation. Unfortunately, the analysts performed so erratically on the task that it was impossible to reach consensus. Templating Guidelines were developed by Sterling Software and counter drug analysts during the early stages of ATS development to define the *proper* templating actions in most cases. Table 2 contains the results of the Sterling Software developers manually templating against the Guidelines, the 7 analysts' results and ATS performance on the same message set.

	Guide	A1	A2	A3	A4	A5	A6	A7	ATS
Ind	97	15	53	83	73	34	72	46	112
Org	7	2	2	5	3	2	4	1	8
Fac	43	1	6	16	5	1	6	12	42
Air	7		1		2		1	5	8
Ves	9		2	11		5	9	3	8
Mat	35	5	6	6	4	6		7	30
Add	34	4	7			2	3	3	34

Table 2: Manual vs. Automatic Templating

The above table shows that ATS matched the Guidelines more closely than the human analysts. In addition, the results for event extraction was even more revealing. The human analysts failed in most cases to identify events at all, while ATS behaved in-line with the Guidelines. In fact, ATS performed consistently better and faster than the analysts in all stages of templating. Several conclusions can be drawn from this test.

Human Analyst Templating Limitations

- Area of Interest - Analysts only care about their particular work domain, so most analysts only extract the information that they feel is important to that domain. Operationally, this means that several analysts must each template the same message in order to extract all of the information.

- Experience/Familiarity with database - Given several options for relating entities, several of the analysts chose very general associations instead of more specific associations or events. Also, analysts rarely noted that entities were related in more than one way. Important information is lost when an analyst selects a **Generic** relationship between an individual and a drug, instead of using the individual and the drug in a **Purchase** event

that includes a time, place, and other information related to the purchase. Experienced analysts performed better than newer analysts in selecting the correct events or relationships.

- Time - Manual templating of a message is a detailed, time-intensive task. On average, an eight to ten sentence message takes about 30 minutes to hand-template and add to the database. Analysts tend to only extract the most obvious items from any message, while ignoring other information that doesn't have any clear importance at that time. The ignored information, though, could be crucial at a later date and therefore still needs to be templated.

ATS Advantages

- More Complete - The system extracts all possible entities, events, and associations and in many cases uses several connections between entities. For any particular entity, there can be an overwhelming amount of detail information included. In addition to an individual's name, for example, the system also can hold physical information (date of birth, height, eye color, tattoos, etc. ...), citizenship and passport information, educational background, as well as several other categories. Although human analysts rarely extract this level of detail, ATS extracts it whenever possible.

- More Consistent - ATS will always template the same information in the same way. This standardization is sorely lacking in the manually-templated data, where a piece of information might be templated in many different ways by different analysts. Greater consistency provides analysts with greater confidence in the quality of the data.

- More Specific - ATS always uses the most specific events and associations available for templating. Some of the analysis tools available in CDIS require very detailed information: for example, the Predictive Analysis tool is driven by event data, which analysts did not template at the same level of accuracy as ATS.

- More Time Efficient - ATS, unlike most analysts, runs 24 hours a day, 365 days a year. In one month of operation, ATS created more than 40,000 useful database records, while only processing one message type. Analysts estimated that this volume of data entry represents the equivalent of 13 analysts dedicated to manual templating over the same period.

5. AUTOMATED TEMPLATING ASSISTANT

Description

Currently, ATS works with four distinct message types. Although this represents a tremendous boon to the analysts, they have in excess of one hundred different message types at their disposal to extract information. Developing systems for each message type would be too expensive in terms of time and computational resources to implement. In order to satisfy analysts' desire for on-the-fly automated templating, it became necessary to develop a generic system that could extract information from any free text message, regardless of message type.

The current state of the art in natural language technology is such that a fully automatic, completely generic data extraction system is out of reach. Perhaps someday, a generic "natural language understanding" program will be written that can perform data extraction, machine translation and speech recognition across all languages, but that day is not in the foreseeable future. Many assumptions were made to make the ATS task manageable.

- Only English texts would be used as input - Any language could have been used, but the initial restriction to only one language allowed more message types to be developed more quickly.

- Separate systems would be developed for different message types - There are significant context clues that can be brought to bear if this assumption is made. For example, in one of the message types handled by ATS, individuals' names are nearly always followed by a date of birth. That context clue is not available in all message types in the counter-narcotics domain, but for this message type, making this assumption improves individual entity extraction significantly.

- Only information that can be added to the database is worth extracting - Only a portion of each message will eventually end up in the database, so by ignoring the rest, the system can process messages more quickly and effectively. For example, materials such as weapons, narcotics and money are all relevant to counter-drug analysis, but references to office furniture are not. The ATS system therefore takes great pains to identify detailed information about weapons or narcotics (type, amount, method of packaging, method of concealment, etc.) , but will not template inventories of office supplies.

The requirement for a generic message templating system forces us to throw out our second assumption, that patterns will be able to take advantage of context clues specific to each message type. Eliminating this assumption has a significant negative impact on the accuracy of the system. The Automated Templating Assistant (ATA) therefore is a generic message templating system which requires human intervention during the data extraction to improve the overall accuracy of the process.

Process Flow

The Automated Templating Assistant performs its tasks in a batch environment, where analysts queue up messages to process, return later to check an intermediate stage of processing, and then finally return to evaluate the results. The specific steps are as follows:

1. The analyst identifies a piece of a message that he/she wishes the system to template. This is accomplished by highlighting the section of text and selecting a menu option to send the text to ATA.

2. ATA receives the text and performs a partial extraction on it, by extracting all of the entities it can. After the entity extraction is complete, the results are saved and made available to the analyst.

3. The analyst checks his/her ATA queue; for each message fragment sent to ATA, there will be a queue entry. The analyst can then view each original message fragment which has been color-coded to identify all of the entities found by ATA. The analyst then marks any entities that were missed by highlighting them and selecting the correct entity type. The analyst can also indicate that an entity was erroneously recognized and tell ATA to undo the identification.

4. Finally, the system processes the message fragment in full once again, using the additional information provided by the analyst in event and association matching; discourse processing, reference resolution and database update all follow as in standard ATS processing.

5. The analyst performs the mandatory verification check on the ATA results and corrects any mistakes. This step is already required for any data entered into the system, whether it originates from ATA, ATS, or another analyst.

Challenges

The key to the success of the Templating Assistant lies in the correction step provided by the analyst. This step replaces the context clues that are lost when the single message-type assumption is removed. By having the analyst identify the type of the entity along with the starting and ending position in the text, the system will be able to perform with a high degree of accuracy. The greatest challenge in developing a more general system, though, remains in identifying entities in a general way. The system will only be a success if the analyst only has to make a few corrections per message fragment. This system is currently under development and will become operational in Fall 1995.

6. TRANSPORTABILITY OF TECHNOLOGY

To this point, ATS has only been applied to processing text in support of the counter-drug community. An important question is, can ATS technology work in any other domain? The answer is simple: ATS can be successful in other arenas, if the following criteria are met:

- Restricted Domain - The broader the domain, the more difficult it becomes for ATS to extract information. There are a limited number of entity and event types of interest to counter-drug analysts; as more entities and events are added to the system, both development time and execution time increases. A relational database usually provides sufficient constraint on the domain for ATS to be effective (though other types of output besides data base records are certainly feasible).

- Consistent Message Types - Consistent message formats and language usage leads to better ATS results. The counter-drug domain involves large numbers of intelligence reports, which tend to be written according to very specific guidelines for content. The language within each message type tends to conform to a particular standard, which can be exploited by ATS. More general purpose types of text, such as news stories, are less well-suited to the fully detailed data extraction which ATS currently provides.

- Reuse of ATS Components - Certain components of ATS are less closely bound to the specific message types or event the counter-narcotics world. For example, ATS currently recognizes every standard (English) means of expressing date-time information; that component could be used across many domains.

- Automated Templating Assistant - Since ATA is not tied to a particular message type, this tool will transport easily into other areas of intelligence analysis.

7. CONCLUSION

Counter drug analysts have at their disposal extensive analysis tools to aid them in their job, but until data is added to the database, these tools have limited effectiveness. ATS frees the analyst from time-consuming data entry and allows them to concentrate on analyzing the data. ATS represents a successful implementation of natural language research and can be applied to domains other than the counter drug domain. Additionally, the Automated Templating Assistant allows the analysts to extract data from any message, regardless of type, expanding the usefulness of ATS even further.

Bibliography

Nancy Chinchor and Beth Sundheim. MUC-5 evaluation metrics. In *Fifth Message Understanding Conference.* ARPA, 1994

Richard Lee, Chris McNeilly, and Louise Osterholtz. The automated templating system (ATS) vs. manual templating. August 1994.

Louise Osterholtz and Mark Holt. The counter drug intelligence system (CDIS). 1995

Lisa Rau and Paul Jacobs. Shogun - system summary. In *Fifth Message Understanding Conference.* ARPA, 1994.

The Authors

Chris McNeilly, Louise Osterholtz and Richard Lee
Sterling Software, ITD
1650 Tysons Boulevard
Suite 800
McLean, Virginia 22102
(703) 506-0800
(703) 506-0154 (fax)

Dr. John Hermansen
Language Analysis Systems, Inc.
Center for Innovative Technology, Suite 201
2214 Rock Hill Road
Herndon, VA 22070
(703) 834-6200
(703) 834-6230 (fax)

Chris McNeilly has worked in the information technology industry for the past 7 years, first as a software engineer for Lexis-Nexis, an electronic publisher that operates search and retrieval systems over enormous amounts of data. He earned a Masters of Science from Carnegie Mellon University in Computational Linguistics in 1992. Since then, Mr. McNeilly has been employed at Sterling Software working on the CDIS project, initially developing a natural language interface to a relational database and then working on the Automated Templating System. He is currently developing the Automated Templating Assistant. He can be reached through email at Chris_McNeilly@sterling.com.

Louise Osterholtz has worked in the natural language processing field since 1990. In 1992, she earned a Masters of Science in Computational Linguistics from Carnegie Mellon University. She has worked on a variety of natural language processing systems, including text indexing and retrieval systems and an English-German-Japanese speech-to-speech translation system. Ms. Osterholtz came to Sterling Software's Information Technology Division in 1993, and since 1994 has worked on the Automated Templating System. She can be contacted at Louise_Osterholtz@sterling.com.

Richard Lee came to Sterling Software in 1992, with 11 years of experience in natural language and information processing technologies. He earned a Masters of Science in Computer Science from the University of Illinois in 1978. Since then he has worked on a variety of natural language processing systems, primarily for the intelligence community, including development of a translator's workstation for non-European languages. Mr. Lee has developed software and knowledge bases for the Automated Templating System, and is currently working on the Automated Templating Assistant. His email address is Richard_Lee@sterling.com.

John Hermansen is the President and founder of Language Analysis Systems, Inc. He received a Ph.D. in Computational Linguistics from Georgetown University in 1985. Dr. Hermansen conducts and directs research and software development in areas of computer science and linguists, principally in problems relevant to name searching and recognition, data extraction and automated data base update. Dr. Hermansen serves as a consultant to Sterling Software for the development of the Automated Templating System.



Linear Scalability on Decision Support Systems: Cray CS6400

Brad Carlile
Cray Research, Inc.
Business Systems Division

1. INTRODUCTION

Decision Support Systems (DSS) manipulate and analyze information to highlight previously unexplored relationships in large Gigabyte- or Terabyte- sized databases. They are emerging as an area of strategic value to many customers in the merchant RDBMS market who need to explore all of their data. In the past, lack of performance has prompted users to sample or to summarize data for DSS processing [1], however, sampling can hide valuable information. Insight can be gained by knowing the detail in the large database that summary data obliterates. Parallel processing enables completely scanning large database for this accurate detailed information in a reasonable time.

New powerful Solaris SMP systems, such as the 64-processor CRAY SUPERSERVER 6400 (CS6400), provide practical platforms that are scalable and flexible enough to handle large databases. The CS6400 is Cray's SPARC-based SMP System that runs the Solaris 2.4 operating system and is fully SPARC binary compatible. The combination of the CS6400 and the parallel features of Oracle7 provide scalable performance for DSS operations. This paper presents results that show near-perfect linear scalability on many of the basic tasks representative of parallel DSS queries (full table scans, nested loop joins, sort merge joins, index creation, etc.) using Oracle7 on the CS6400. These basic tasks are the components of more complex queries. Optimization of these components improves "real world" user performance. The key to high delivered performance on DSS is taking advantage of the important application characteristics and focusing on the important system parameters in the complete system.

2. DSS CHARACTERISTICS

An understanding of an application's characteristics is important when implementing a balanced system. Several characteristics of DSS dictate the need for different tuning strategies than are applied to traditional OLTP applications. Important aspects of DSS operation are the ad hoc nature of the queries, the parallelism that can be applied to queries and the data movement patterns within the hardware system. These aspects, when properly measured, can shed additional information on obtaining high performance on DSS operations.

The nature of DSS systems is to iteratively refine and define new queries based on the information gathered from previous queries. These queries are ad hoc and unpredictable. It is difficult to preplan for these types of queries since they are only executed once and a query may access millions or billions of rows [1]. With ad hoc queries there is no perfect data layout, especially when refreshing the database with inserts and updates imbalances the original data layout. For predictable performance on SMP systems, fine-grain distribution of data evenly across the disks provides equal access time to the data. Without this equal time access of data, bottlenecks can degrade performance by orders of magnitude and serialize the processing. Such performance problems are a defining characteristic of MPPs [6] [8]. Alternatively, high-performance SMP systems are very flexible and capable.

A characteristic of DSS applications that take advantage of parallel processing is the ability to divide a single query into sub-queries. Executing in parallel keeps both processors and disks active reducing execution time. In Oracle, these sub-queries execute on multiple "Query Servers" in parallel and provide results to a Query coordinator that combines results as required by the query. Parallelism,

such as on the CS6400, provides a cost-effective approach to meeting typical DSS performance requirements on large databases.

Internal and external data movement is critical to performance and is often an overlooked characteristic of RDBMS operation. This is becoming more critical as the gap between processor speed, memory speed, and disk speed grows [3]. Optimal disk reference patterns are quite different in OLTP applications then they are in DSS applications. Typical OLTP disk accesses are totally random and are typically in the 2 Kbyte to 4 Kbyte size. The majority of OLTP transactions only need data from only a few rows in a few tables. The appropriate performance metric for these small reads and writes is IOs/second. In contrast, many important operations in the DSS applications tend to read many consecutive rows of a particular table (table scans, aggregates, group-bys, joins, etc.). To optimize IO for DSS, the disk accesses issued by the RDBMS should be very large, up to 1 Mbyte or more, and consecutive. Under these characteristics, the appropriate performance metric for these large reads is Mbytes/second. Within a single RDBMS, it is possible to tune it to implement OLTP transactions with small-sized IOs and to implement DSS-style queries with large-sized IOs. For instance in the Oracle RDBMS, the "db_file_multiblock_read_count" parameter allows the DBA to set a larger read size for DSS style queries. Currently, we believe that delivered application disk performance for large DSS databases should be on the order of the hundreds of Mbytes/sec.

3. Measuring DSS Performance

During the tuning process it is necessary to establish the characteristics of a DSS application by measuring its performance. Appropriate performance metrics measure the important characteristics of a particular operation. In addition, they provide a reasonable prediction of performance when changing the dimensions of the database. Appropriate measures of DSS performance are MB/sec delivered during a query and the percentage of the job that is parallel. Some performance metrics do not allow accurate comparisons between different implementations and should not be used to predict performance.

The MB/s figure for a particular query will be a good predictor of the performance since a portion of most DSS queries consist of large consecutive IO operations. During these operations, entire rows move from disk to memory even when accessing a particular column of the row. The best characterization is the time it takes to move this data from disk and process it (MB/s = size of the table/time to scan the table). This disk transfer time typically dominates the computation. Given a particular query type, MB/s will be roughly constant on tables of different sizes.

An inappropriate performance metric is millions of rows/second. The problem with this measure is that the size of a row is highly dependent on the table design (a row may contain tens, hundreds, or thousands of bytes of data). For different size tables, full table scan times may be very constant in terms of Mbytes/sec whereas Mrows/sec varies by almost 3 orders of magnitude as is illustrated in the table shown below. Mrows/sec is an inappropriate performance metric given the inherent variability in row size.

Comparison of MRows/Sec and Mbytes/sec

Rows	Bytes/Row	Seconds	Rows/Sec	Mbytes/Sec
2,000,000	2000	66	0.03 Mrows/s	61 MB/s
5,000,000	200	16	0.30 Mrows/s	63 MB/s
10,000,000	50	9	1.20 Mrows/s	57 MB/s
200,000,000	150	470	0.42 Mrows/s	64 MB/s

On parallel systems, a useful DSS performance metric is scalability. Caution should be exercised when discussing scalability. Scalability figures can be artificially inflated by crippling single processor performance and optimizing parallel performance. It is important to test the application with the appropriate tuning parameters for both parallel and sequential executions. It is best to be

suspicious of scalability calculations based on best parallel runs against initial (un-tuned) single processor runs.

We suggest that the best manner to look at scalability is the speedup on a particular number of processors or the percentage of parallelism. The only manner to accurately compare system scalability is to use actual performance. The maximum number of processors on a system does not determine its scalability. The percentage of an application that is parallel and the overheads involved in using multiple processors limit delivered parallel performance. Parallel performance beyond a given number of processors can be estimated using a formula based on Amdahl's law [3]. This estimate involves determining the percentage of a job that is parallel and predicts the speedup for a given processor count using "percent parallel".

To determine the percentage of a job that is parallel (P= percent parallel), a one-processor run and a 40-processor run (full table scan with aggregates) will be used to calculate percent parallel.

$$P = (1/observed_speedup-1)/(1/n-1) \qquad (1)$$

where n is the number of processors and speedup is the observed speedup. For example, if a 40 processor can get a 39.083x speedup over one processor, then the percent parallel is: P = (1/39.083-1)/(1/40-1) P = .99939 or 99.939% parallel. To predict other speedups, use the following formula:

$$predicted_speedup = 1/(P/n + (1-P)) \qquad (2)$$

where P is the percent parallel and n is the number of processors. Using the example above (P=.99939), we get the following table, which shows good agreement with the actual results.

Prediction using Amdahl's Law			
N	Calculation	Predicted Speedup	Actual Speedup
1	*actual data used*	1.0	1.0
8	1/(.99939/8+(1-.99939))	8.0	8.7
16	1/(.99939/16+(1-.99939))	15.8	14.4
24	1/(.99939/24+(1-.99939))	23.7	23.4
32	1/(.99939/32+(1-.99939))	31.4	32.3
40	*actual data used*	39.1	39.1
56	1/(.99939/56+(1-.99939))	54.2	Est.
64	1/(.99939/64+(1-.99939))	61.6	Est.

Potential errors in prediction may arise from the following areas:

- Performance limitations due to application characteristics or by the IO or memory bandwidth of the system.
- Changing the size of a problem will usually increase the time spent in a parallel region (it is very difficult to use the above estimation for a different problem size).
- Estimates can be very low if more parallelism exists by tuning the code (improving performance of serial section or making more of it parallel).
- Estimates can be very low if using more processors increases the percentage of a job that is parallel. (cache effects such as interference and memory layout).

In the above example, the percent of parallelism was a good predictor of performance. The different DSS queries measured in this report are approximately between 98.00% and 99.93% parallel.

Another performance metric used by some is percent speedup (*actual speed/perfect speedup*). This is not an accurate manner to report scalability since it varies with the number of processors. In the table above, the percent parallel was roughly constant. If we looked at the erroneous percent speedup, we would see that this figure varied between 100% and 97% and this figure will decrease as the number of CPUs grow. This measure does not give an accurate view of the performance and should not be used.

4. IMPORTANT SYSTEM PARAMETERS

A DSS system consists of many components. With respect to performance, the order of importance of these components is the RDBMS, operating system, delivered disk bandwidth, delivered memory bandwidth, and finally processor speed. There are many interdependencies between various aspects of these performance components.

RDBMS

The implementation of the RDBMS is critical for database performance. This is especially critical for parallel-processing performance. In general, queries can be decomposed to run in parallel in a variety of ways. A query optimizer needs to be intelligent enough to determine an appropriate parallel query plan of execution. It should effectively choose between options such as using table scan versus index reads or a particular implementation of table joins. An efficient implementation balances query decomposition for the system configuration and database table sizes. During query execution, multiple processes need to be efficiently used and coordinated. In addition, the RDBMS needs to be optimized for memory management and to properly utilize its cache (System Global Area or SGA). Issues related to RDBMS performance include shared resource utilization, locking issues, IO strategies, and efficient code. Oracle incorporates all of these features in a shared-everything architecture.

Important tunable Oracle parameters are located in the *init.ora* file. Parameters of interest to DSS workloads involve SGA size (up to 2 GBytes), database block size, multi-block read size, query server sort area size (not limited by SGA size), and the number of query servers. The DSS Benchmark section below discusses some of these in more detail.

Operating System

The operating system (OS) can limit the efficiency and scalability of RDBMS performance. Efforts by Cray and Sun [4] [2] allow the RDBMS to exploit performance features of Solaris 2, such as the multi-threaded architecture of the Solaris kernel, asynchronous IO, soft processor affinity, efficient OS striping, and enhancements to memory management for large memory systems. The combination of these efforts and others provides a system that is responsive and efficient when executing parallel workloads.

There are very few tunable OS parameters since the operating system is tuned for RDBMs workloads. Parameters that may be of interest to some workloads are scheduler classes, increased semaphore limits, and the file system flusher. Providing application scalability has been a design requirement of Solaris for many years.

Disk Bandwidth

The CS6400 has delivered over 265 MB/s on a moderate size disk configuration (90 Elite3 disks). This is more than some vendor's memory bandwidth. The program tested in this case issued continuous reads with no computations. This characterizes the maximum realized disk bandwidth for this configuration. IO-bound applications are likely to deliver less than this figure due to additional required application processing. Delivered disk performance on a particular disk configuration provides a much better baseline for performance estimation than system maximums listed on spec-sheets.

In order to obtain sufficient I/O for DSS applications, disk reads must be of a sufficient size to maximize bandwidth as opposed to the small I/O's typically used for OLTP applications. A simple manner to optimize the performance of large IOs is to use a Volume Manager (Solstice DiskSuite, Veritas Volume Manager, etc.) to stripe the data across the disks. Disk striping can also be viewed as a way to optimize data layout on disks. The fine-interleaving of datablocks across the disks has the

advantage of naturally distributing inserts and updated data throughout the disk system. Ad hoc queries are naturally optimized. There is no processor dependence on data layout, simplifying performance tuning.

Fine-grain disk striping (64k to 1M) can increase disk bandwidth and minimize disk hot-spotting. Fine-grain striping is most generally applicable to a wide range of query types. Alternatively, course-grain disk striping (concatenated disks) may give higher performance for only certain queries. For example, the CS6400 with a moderate size disk system has delivered over 110 MB/s on a disk-resident database using Oracle.

Memory Bandwidth

Memory Bandwidth of a system is also a major contributor to system cost. Efficient use of the available bandwidth is critical. The CS6400 delivers over 1100 MB/s of memory bandwidth. Delivered bandwidth for cached data is much higher.

Memory size can also be an issue for DSS performance. Data that does not fit in the physical memory of the system will reside on the disk. The CS6400 supports up to 16 GBytes of physical memory. This allows for a large SGA and large sorting area. Swapping and paging are not generally an issue with the large configurations of the CS6400. The large memory also provides the ability to cache indexes or other randomly read "hot" tablespaces. One manner to cache these tablespaces is to put them in a Unix File System (UFS) and let the Unix file caching mechanism buffer data from these tables. Using this method, it is possible to effectively cache randomly accessed tables that approach physical memory size. This has increased performance for some workloads.

Processor Speed

Processor performance can be measured by using the CPU speed or by using "cache-friendly" benchmarks such as SPECint92. Processors designed to deliver high SPECint92 performance focus primarily on processor cache-to-processor issues. These benchmarks do not even stress the processor-to-memory issues that are important to most applications. For these reasons, SPECint92 results can be very misleading when they are used to gauge performance of "non cache friendly" operations more typical in RDBMS code. DSS performance is more likely to be determined by IO speed or the efficient implementation of the RDBMS.

Processor speed is important, but it must be balanced with processor-to-memory bandwidth and efficient IO. These are factors that SPECint92 does not measure. Recent data [5] [7] suggests that delivered MB/s is a better estimate than SPECint92 for DSS performance. Even though this data is not directly comparable, it can be instructive. The table below shows that a DEC 7000 (275 MHz) has a much higher SPECint92 rate than the CS6400 (60 MHz) but has a much lower delivered performance on DSS operations. Results later in the paper show that the CS6400 performance will scale with more processors. Digital has not published any data on parallel DSS performance.

	CS6400 (1 processor) [5] SPECint92 = 89	DEC 7000 (1 processor) [7] SPECin92 = 180
Full Table Scan	4.1 MB/s	3.4 MB/s
Multi-Table Join	1.3 MB/s	0.3 MB/s
Index Creation	0.9 MB/s	0.1 MB/s

As shown above, the CS6400 is 1.2x to 9x faster than the DEC 7000 on DSS operations, however this would not be predicted by the SPECint92 rating (or CPU MHz). SPECint92 should not be used to predict DSS performance.

In addition, Stephen Brobst [1] has proposed a constant based on SPECint92 to estimate performance that a processor should have for DSS operations. His constant is (10 SPECint92/Disk spindle). This number comes from experience on the Teradata system (486, SPECint92 = 32) which gets maxed out at 3 disks or 10 (SPECint92/Disk spindle). Due to the variability of this constant on different systems

and its reliance on SPECint92, it is not recommended that this measure be used to compare systems or estimate requirements for a balanced system.

5. DSS BENCHMARK

To demonstrate the high scalability that Oracle provides on powerful SMP systems, several queries representative of important DSS operations were executed with a CS6400 system. The benchmark consisted of tuning the database for optimal performance and measuring various system functions and scalability with different numbers of processors. The CS6400's configuration consisted of forty 60 MHz SuperSPARC processors, 1240 MB of physical memory, and ninety (2.9 GB) disks. Oracle version 7.2.1 was used together with Sun's Online DiskSuite Version 4.0.1 (beta) which provided perform machine (OS) striping for the data files. The data tablespace was evenly spread out across 72 disks using a stripe size of 64K. The SORT_AREA_SIZE was 20 MB per query server process, DB_BLOCK_SIZE was 8K, and the number of DB_BLOCK_BUFFERS was 6400 (approximately 52 MBytes). The data tablespace had a PCTINCREASE of zero. The tuning process only involved adjusting Oracle init.ora parameters.

A 5 million row table with 16 columns and a 204 byte average row length was used to test the Oracle7 parallel features. A single Oracle instance was used with a single user issuing the queries sequentially. The amount of table data was constant during the execution of the following tests:

Table Scan with Aggregates (SCAN). This operation scans the table and performs an aggregation on each of the sixteen columns in the table. Aggregation functions used were min, max, avg, sum, and count.

Sort-Merge Join (SMJ). This entails joining the table to itself. Here the table is scanned twice where each scan chooses half the number of rows resulting in 5 million rows being grouped by different criteria, resulting in 1,000 evenly distributed groups that are ordered. The sort-merge join is selected over nested-loop join by the use of an optimizer hint.

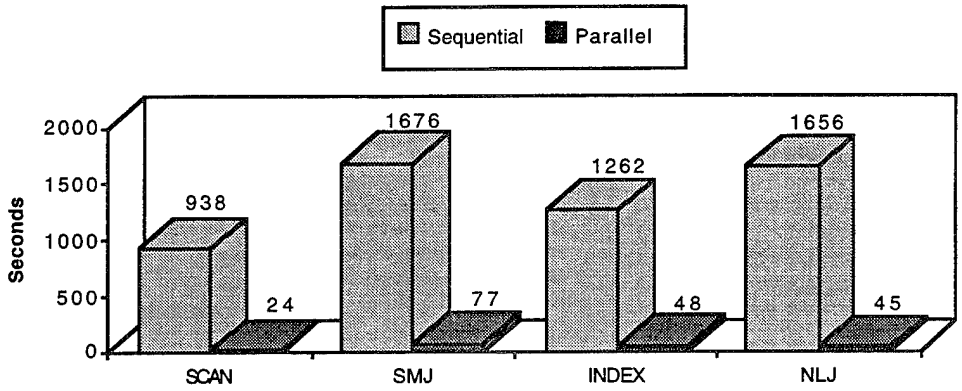
Parallel Index Creation (INDEX). For this operation, a set of query servers scans the target table, and passes the row IDs and column values to another set of query servers that sorts the index entries. These sorted entries are finally passed to the query coordinator process that builds the index. For this test an index was built on a single numeric column in the table.

Nested-Loop Join (NLJ). This operation scans the table with 2% of the rows selected to join with another instance of the same table. The join is 1:1 so 100,000 index lookups are done on the index created by the index creation test. The joined rows are then filtered by the predicate in the inner table returning 10 rows. The nested-loop join is selected over sort-merge join by the use of an optimizer hint.

Results

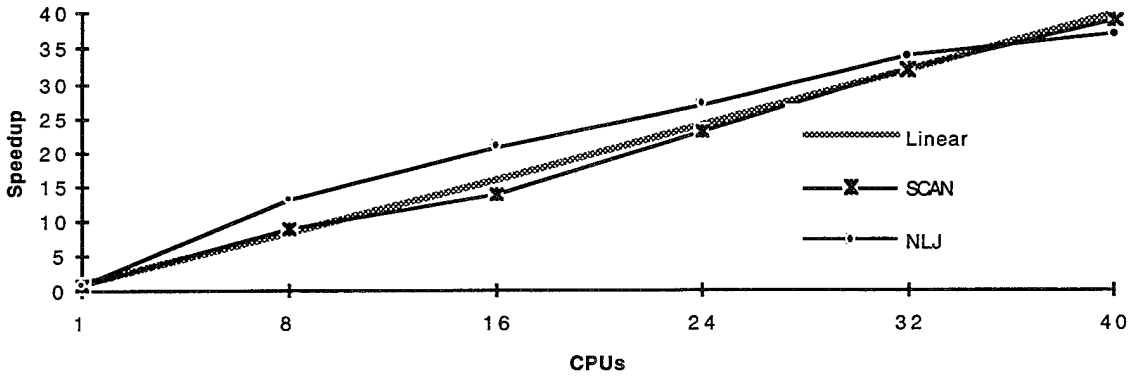
The SCAN operations using 40 processors took only 24 seconds to complete while one processor took approximately 16 minutes to complete. This is a speedup of 39 times compared to a non-parallel operation with a single CPU. The NLJ operation showed a 37x speedup, reducing the query time from half an hour to 45 seconds. The speedup for the SMJ operation is also significant since the initial scan as well as the subsequent join and group-by operations are parallelized. In each of these cases, all processors are effectively used. Even more performance is expected with an expanded IO system with more disks and controllers to increase the delivered IO bandwidth for these IO intensive operations.

Sequential vs Parallel Performance on Cray CS6400 (40 CPUs)



These benchmark results demonstrate near-linear scalability. Analysis of these queries show an average of 99% parallel, which as discussed above, predicts that all 64 processors can be effectively used on the CRAY CS6400 system. The graph below compares actual results with perfect linear scaling with the number of processors. During these runs, only the number of query servers was varied with other parameters kept constant. Optimal performance was obtained at 1 processor and 40 processors, not optimizing the performance for other processor configurations.

Linear Scalability on Cray CS6400 (40 CPUs)



Another feature explored is parallel loading of the database. For large databases this can be an important time-consuming operation. Multiple processors will greatly accelerate this operation. Parallel Load involves reading multiple data files from disk and using the multiple concurrent direct loader sessions to write this data simultaneously to the same table in the database. Performance on Parallel Load is IO bound and additional performance improvement can be obtained by increasing the number of disks and controllers.

Updated and additional results will be made available at the conference presentation.

6. CRAY SUPERSERVER 6400 SYSTEM

The CS6400 is an enterprise-class application or data server for a wide range of tasks such as on-line transaction processing (OLTP), decision support systems (DSS), on-line analytical processing (OLAP), or data warehousing. The result of a technology agreement between Cray Research and Sun

Microsystems, the CRAY CS6400 is a binary-compatible upward extension of Sun Microsystems' product line. Its full compatibility with Sun Microsystems' Solaris operating system guarantees the availability of the largest set of third-party solutions in open systems. Large configurations of this SMP system can simultaneously support sixty-four processors, 16 Gigabytes of physical memory, and 10 terabytes of online disk storage. The CS6400 also has the capacity to combine DSS and online transaction processing (OLTP) job mixes on the same platform. The CS6400 also provides processor partitioning to segregate these workloads for flexibility in system management. In addition to DSS scalability, the CS6400 has also shown excellent OLTP Scalability. It leads in the industry in TPC Benchmark™ B Results with a performance of 2025.20 tpsB and leads in price/performance with $1,110.14 per tpsB (result date: 6/4/94).

RAS features are a critical part of the design of the CS6400. There is nearly complete redundancy of system components in the CS6400. This includes multiple redundant system buses, N+1 power supplies, dual pathing, RAID devices, disk mirroring, etc. The CS6400 also offers fail-over, hot swap of system boards, dynamic reconfiguration (and expansion), and automatic reboot. A separate service processor including monitoring software (with call home on unplanned reboots) and remote diagnostics.

The speedup factors obtained are the result of joint engineering efforts by Oracle, Cray, and Sun in exploiting the performance features of Solaris 2, such as the multi-threaded architecture of the Solaris kernel, asynchronous I/O, and efficient OS striping. Likewise, the hardware strengths of the CRAY SUPERSERVER 6400 that facilitate good scalability include the quad XDBus bus architecture, fast SCSI controllers, and larger CPU caches to hold frequently referenced data and instructions. Oracle will exploit faster CPUs with larger caches to deliver even bigger performance boosts for future generations.

The SMP architecture allows DSS queries to be optimized for parallel operations, while avoiding the MPP performance and administration problems. MPP performance can be very dependent on data layout. On MPPs, the user has the choice between executing high-performing "good" queries and slow-performing "bad" queries. This has the drawback of potentially "training" users what queries not to submit. In these respects, MPPs are more difficult to tune and to administer. Even on an MPP that uses a shared disk strategy, there can be other problems on an MPP due to coordinating the various IO requests from within the MPP.

System Components	Configurations	Specifications
Number of Processors	4-64 SPARC	60 MHz SuperSPARC
Memory Size	16 Gbytes	SMP, Shared Memory
System Bandwidth	1.7 GB, 4 XDBuses	55 MHz
I/O Channels	16 SBuses	800 MB/s
Bus Controllers	64	Full Coherency
Online Disk Capacity	10 Tbytes	Using 9 GB disks
Operating System	Solaris 2.4	SVR4, Solaris Enterprise Server

7. CONCLUSION

Performance and scalability are particularly important for DSS applications. The CS6400's SMP design allows commercial DBMSs to effectively use its large configuration of processors. Large configurations of the CS6400 provide excellent scalability on DSS operations using the Oracle7 shared-everything implementation. The characteristics of DSS operations allow IO optimizations that deliver high bandwidth. The efficient implementation of the RDBMS on the C6400 provides near-linear scalability while maintaining all of the advantages of SMP systems. These effects are effectively demonstrated using the MB/s and percent parallel metrics. Past limits to SMP scalability are avoided by providing sufficient performance at every level in a balanced system.

8. REFERENCES

[1] S. Brobst, "An Introduction to Parallel Database Technology", VLDB Summit, Miller Freeman, Inc, 1995.

[2] A. Cockcroft, Sun Performance and Tuning, SunSoft Press, A Prentice Hall Title, 1995.

[3] J.L. Hennessy and D. A. Patterson, *Computer Architecture A Quantitative Approach* (Morgan Kaufmann Publishers, San Mateo CA, 1990).

[4] D. McCrocklin, "Scaling Solaris for Enterprise Computing", Cray User's Group, 1995.

[5] *Oracle and Cray Superserver 6400 Linear Scalability*, Oracle Corporation, May 1995.

[6] "Open Computing & Server Strategies", Fourth Quarter Trend Teleconference Transcript, META Group, Dec 13, 1994.

[7] J. Scroggin, "Oracle7 64-Bit VLM Capability Makes Digital Unix Transactions Blazingly Fast", Oracle Magazine, Vol IX, No 4, July/August 1995, pp 89-91.

[8] C. Stedman, "What you don't know... ... will hurt you", Computer World MPP & SMP special Report, March 27, 1995, supplement pp 4-9.

9. AUTHOR INFORMATION

Speaker's Biographical Sketch

Brad Carlile is a Performance Analyst at Cray Research, Business Systems Division. He is responsible for analyzing and characterizing real-world workloads. Background includes work on eight distinct shared and distributed memory parallel architectures on a wide variety of commercial and technical applications. His current focus is DSS performance issues.

Contact Information

Brad Carlile
Cray Research, Business Systems Division
8300 Creekside Ave,
Beaverton, OR 97008

bradc@oregon.cray.com
(503)520-7622 (voice)
(503)520-7724 (fax)

Managing Interface Migration in DOD

M. Cassandra Smith
Susan L. Ficklin
Darin S. Satterthwaite
David J. Connolly

The MITRE Corporation

In a memorandum to Department of Defense (DOD) managers, the Assistant Secretary of Defense (ASD) for Command, Control, Communications, and Intelligence (C3I) instructed owners of systems or applications in DOD to record sources (other systems or applications) that send them data. This data represents an interface between the receiving application and the source application. The purpose of recording and registering these interfaces, according to the ASD C3I, is to ensure that needed system interfaces and data exchanges are not mistakenly eliminated or disconnected when systems are migrated into the DOD target environment.

MITRE has participated in a project to 1) define a process for managing the interfaces, which are cross-functional and cross-Service; 2) develop a process to evaluate and select an automated tool to manage the interfaces and provide analytical support; and 3) define a data model to support the process to manage interfaces and to serve as the underlying data model for the automated tool. This abstract reports on the work MITRE performed on the project, which supports the Defense Information Systems Agency (DISA).

MITRE, in collaboration with DISA and other users, developed an IDEF0 process model to define the process for managing interfaces. Figure 1 provides an overview of the first level of the model, which is named "Manage Interface Migration." As the figure shows, the process consists of four basic activities: 1) Establish Direction; 2) Maintain Interface Architectures; 3) Manage Interfaces; and 4) Identify Opportunities and Implement Recommendations.

Figure 2 is also an IDEF0 model that depicts the steps involved in evaluating and selecting a tool for the Manage Interface Migration process. An important result of the activity was the creation of an extensive list of functional requirements using the Manage Interface Migration process and data models as a basis.

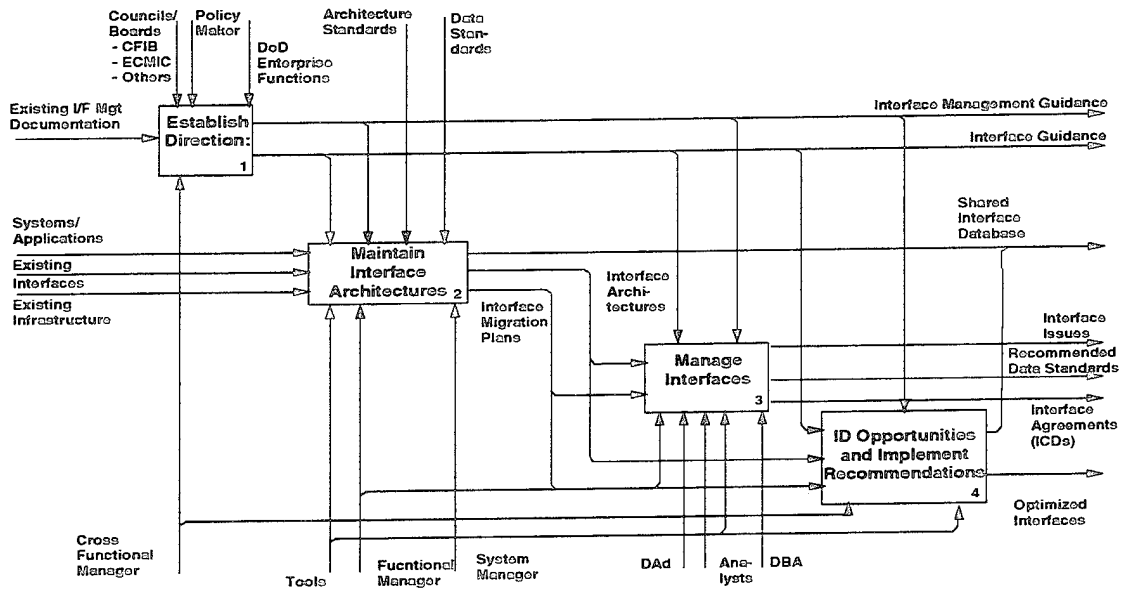


Figure 1. Manage Interface Migration Overview

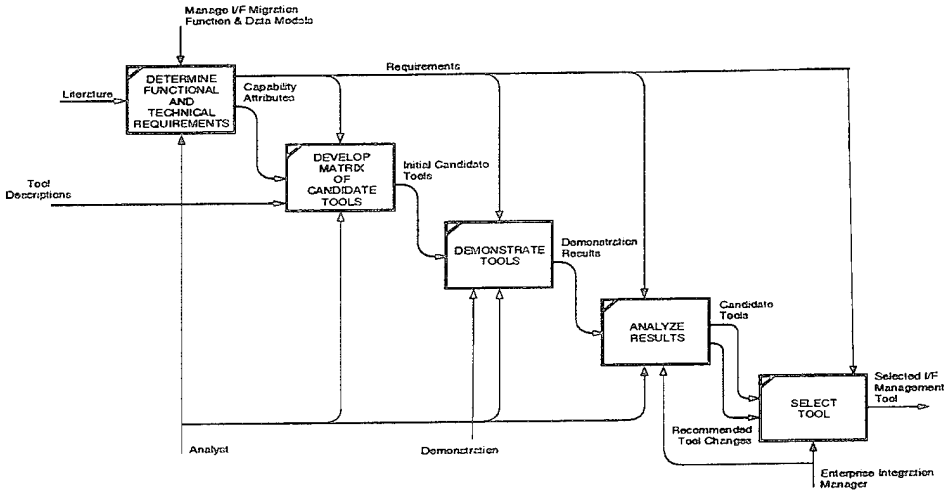


Figure 2. Manage Interface Migration Tool Assessment Process

Figure 3 shows the conceptual data model for the Manage Interface Migration activity. The APPLICATION INTERFACE entity is the focal point of the model. The important relationships are those between the APPLICATION INTERFACE entity and the APPLICATION entity, which indicate that applications interface with other applications to share data. MITRE also developed a fully attributed and normalized logical data model in IDEF1X.

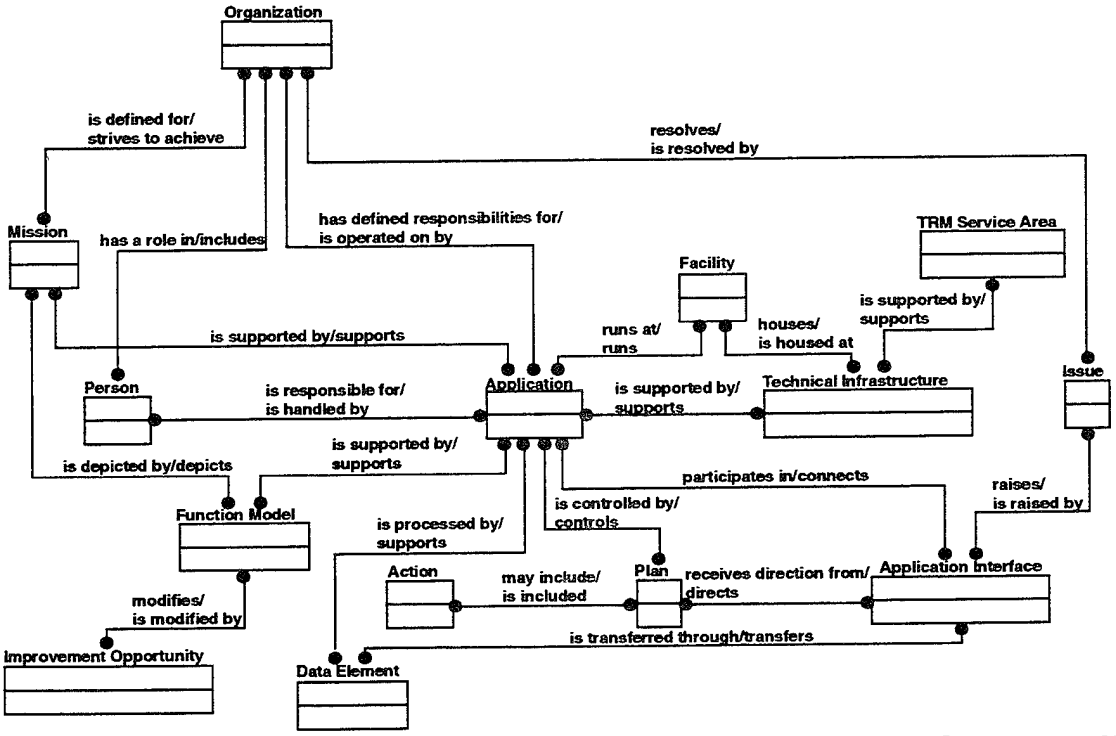


Figure 3. Manage Interface Migration Conceptual Data Model

The following summarizes the current status of the project:

- Twelve percent of required data has been collected. In conjunction with this, 1) needed legacy functions that will migrate to the new DOD environment have been identified; 2) a standard data collection method is in place; and 3) migration systems have recorded their interface information.
- Clear, continuing communication exists between the Office of the Secretary of Defense (OSD), DISA, the DOD functional areas, and the Services.
- Tools, which are widely available and easy to use, are under evaluation.

The next steps are:

- Validate the function and data models and integrate with the DOD enterprise strategy for shared data.
- Select the automated tool(s).
- Specify the schedule.
- Promulgate the common policies and procedures for managing interfaces across DOD.

Biographical Sketches of Authors:

Cassandra Smith is a member of the technical staff at the MITRE Corporation in the Enterprise Engineering department. Her interests include database management systems (DBMSs), especially object-oriented DBMSs, and systems design. She has a B.A. degree from Howard University and M.S. and Ph.D. degrees in computational linguistics from Georgetown University. Cassandra can be reached at The MITRE Corporation, Mail Stop W644, 7525 Colshire Drive, McLean VA 22102. mcsmith@mitre.org, (703) 883-6703 voice, (703) 883-3383 fax.

Susan Ficklin is a group leader at the MITRE Corporation in the Enterprise Engineering department. Her interests include information systems architectures and enterprise integration. She received a B.A. degree from Rice University and an M.S. degree in Systems Engineering from George Mason University. Susan can be reached at The MITRE Corporation, Mail Stop W644, 7525 Colshire Drive, McLean VA 22102. sficklin@mitre.org,(703) 883-6075 voice, (703) 883-3383 fax.

Darin Satterthwaite is a Member of the Technical Staff in The MITRE Corporation's Washington, D.C. Software Engineering Center, Information Systems Department. He has experience supporting a variety of tasks in the Systems Development Life Cycle, ranging from requirements analysis to database support. Darin holds a B.S. degree in Business Administration with a concentration in Information Resource Management and an M.S. in Information Systems, both from George Mason University. Mailing address: The MITRE Corporation, Mail Stop Z667, 7525 Colshire Drive, McLean VA 22102-3481. Phone: (703) 883-6638. Fax: (703) 883-6991. Email: dsatter@mitre.org.

Dave Connolly is a department assistant in the Enterprise Engineering department of the MITRE Corporation. He joined MITRE in 1990 after serving in the U.S. Coast Guard since 1963. Education and training conducted during his Coast Guard service included: Naval War College, U.S. Naval Postgraduate School (Masters-Systems Analysis), U.S. Naval Flight Training (aviator), and USCG Academy. He can be reached at the MITRE Corporation, Mail Stop W644, 7525 Colshire Drive, McLean, VA 22102. dconn@mitre.org, (703) 883-5429 voice,(703) 883-3383 fax.

PITFALLS, TRAPS AND GOTCHAS IN BUILDING A LARGE, MULTI-LEVEL SECURE DATABASE

Mike Lefler
PRC Inc.
1500 PRC Drive
MS: 5S2A
McLean, VA 22102
(703) 556-1863

James Bradley
Coleman Research Corp.
9891 Broken Land Parkway
Suite 200
Columbia, MD 21046
(410) 691-5242

Trusted components do not a trusted system make. The *Department of Defense Trusted Computer Security Evaluation Criteria* manual (TCSEC or "Orange Book") consolidates knowledge about the degree of trust one can place in a system designed to protect sensitive information, but in practice a number of design problems arise which are specifically related to information security where the system developers are essentially on their own. This presentation draws upon actual experience developing an multi-level secure (MLS) information system to illustrate the types of security-related design problems which can arise above and beyond the normal database implementation issues. Security requirements had a major impact on the design of the software processes for this system, and while some of these are specific to the system under development and its environment, others are generic and represent problems likely to recur in other MLS system development efforts.

1. BACKGROUND

For the past three years PRC has been developing a large information system for a Department of Defense (DoD) agency under the cover name "ENGRAFT." ENGRAFT is interesting in that it combines aspects of on-line transaction processing (OLTP) with data warehousing in a distributed, high availability, MLS environment. As might be imagined, this program pushed against the cutting edge of technology in several simultaneous directions, with MLS being one of the more challenging facets of the problem.

ENGRAFT will replace a legacy system developed some number of years ago, which was implemented in COBOL using a proprietary CODASYL database management system. The legacy system provided a measure of information security by attaching a numeric string to records in the database to serve as a mask to determine which of the

connected terminals the data could be displayed on. Such an ad hoc approach was unacceptable for ENGRAFT, which is required to make the data available to end users via ad hoc query and reporting tools (in our case, DataBrowser). Moreover, difficulties in maintaining the legacy system's COBOL code made the Government sensitive to the advantages of using commercial front end tools to create a graphical user interface (GUI) in an open, client-server environment.

2. PROBLEMS ENCOUNTERED

One of the largest headaches building any replacement information system is cross-loading the old data to the new. In ENGRAFT we experienced the standard difficulties, such as dates in the wrong format (DDMMYY instead of YYMMDD), creative reuse of obsolete or empty fields by the user community, and so forth. On top of this we were also faced with the problem of correctly labeling the legacy data. The data in ENGRAFT is highly compartmented, and setting the correct compartment, handling caveats, and country releasability for several million rows of data could not possibly be performed by hand. Attempts to develop algorithms to deduce compartments, caveats, and releasability from the legacy system's terminal access control mask were not successful.

Related to the problem of labeling legacy data is the problem of labeling new data. ENGRAFT receives most of its data from other agency information systems. These systems are not MLS, and consequently the data enters ENGRAFT in an unlabeled state. Moreover some reports will reach ENGRAFT directly in electronic form, and must be parsed by ENGRAFT itself. Regardless of the source, ENGRAFT must validate the data, correlate it to existing information, correctly label it, and pass it to Information Management Officers for quality assurance before it becomes available to end users to support research and analysis as per the flow depicted in Figure 1.

Security labels have two parts: a hierarchical *classification level*, such as CONFIDENTIAL or TOP SECRET, and a non-hierarchical collection of categories or *compartments*. When a user connects to Oracle a security label is assigned to the user's session, and the user may only access data which is dominated by the user's session label. That is, the classification level of the data must be less than or equal to the classification level of the user's session label, and the data's compartments must be a subset of the compartment component of the user's session level. The large number of workstations which will ultimately connect to ENGRAFT, coupled with the high cost of MLS operating systems, resulted in an ENGRAFT requirement that it correctly establish a user session label for users accessing the system from untrusted (i.e., not running an MLS operating system) workstations.

At first blush this requirement seemed straightforward to handle, since Trusted ORACLE7.1 allows the assignment of default session labels to workstations running an untrusted operating system. But labeling a workstation to reflect the clearances of its owner had a serious hole -- if user B logs onto ENGRAFT from user A's workstation with their own user identifier and password, then user B receives his or her own discretionary access permissions but receives the default session label of user A's workstation, i.e., he or she would inherit user A's security clearance. A suggestion that

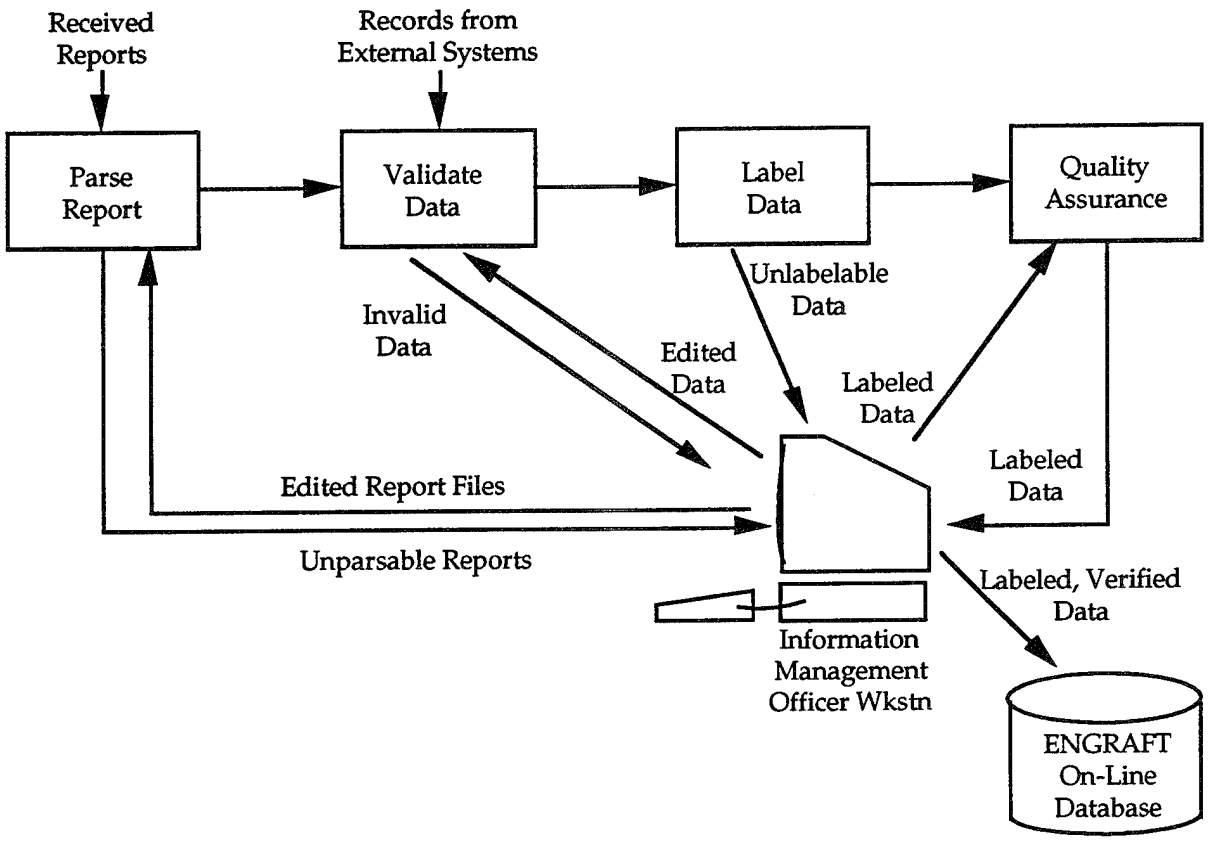


Figure 1: Information Flow for Receipt of ENGRAFT Data

the workstations be physical secured against this possibility did not fly with the government program management team at all. Somehow the ENGRAFT login process for untrusted workstations would have to cause Trusted ORACLE7.1 to assign a session level which was the greatest lower bound (GLB) of the maximum security level allowed for the user and the single level assigned to the workstation.

Finally, ENGRAFT has requirements which allow users at remote, non-US sites, to have interactive access to data. These users may be either US citizens assigned outside the United States, or foreign nationals who have been granted access. However users at non-US sites, whether US nationals or foreign nationals, may only have access to that portion of the ENGRAFT database which has been specifically marked as releasable to the host country. Since Trusted ORACLE7.1 delegates basic security labeling and mandatory access control (MAC) enforcement to its underlying trusted operating system, and since Data General's B2 DG/UX operating system does not support country releasability or handling caveats, this was one of the more challenging issues addressed by the ENGRAFT development team.

3. SOLUTIONS

The data load problem was solved with the assistance of the government with the development of an algorithm to label the legacy data according to the values of certain

fields. Serendipitously, the same labeling algorithm can and will be used to label incoming data received electronically. This algorithm has been tested and validated against the legacy data and samples of the data to be received from the currently-existing systems.

The way ENGRAFT establishes an end user's session label involves the creation a special Oracle role (named MACPRIVS) which has READUP, WRITEUP, and WRITEDOWN privileges, and the creation of two tables to store user clearances for all authorized user identifiers and default workstation session labels for all workstations, respectively. The procedure for connecting to ENGRAFT is as follows:

1. The user logs onto the (untrusted) workstation

2. The user pulls up the ENGRAFT login window and enters the user identifier and password.

3. Special ENGRAFT application software connects to Oracle with the userid and password, and sets the user's role to MACPRIVS using a carefully safeguarded role password.

4. The application software retrieves the user's clearances and the workstation default session label, and calculates the intersection of the set of compartments and the lesser of the two hierarchical classification levels using the Oracle GLB function.

5. The user's session label is altered to the session label calculated in Step 4 and the MACPRIVS role is exited.

The software which implements this process has been subjected to rigorous scrutiny, and the procedures for safeguarding access to the database tables and the MACPRIVS password have likewise been carefully evaluated.

Solving the problem of country releasability required the ENGRAFT development team to re-map the DG/UX sensitivity label. The current DG/UX sensitivity label consists of a one byte classification level and 128 bits of compartments/categories. This sensitivity label structure provides 250 more classification levels than are needed to cover the range of data security classifications stored within ENGRAFT, but ENGRAFT is contractually required to provide a minimum of 64 compartments so there are only 64 additional bits which can be used to support caveats and releasability. Although the total number of countries in the world is quite large (and growing almost daily), the number of countries which will have sites which may connect to ENGRAFT is substantially smaller, so it is reasonable to set aside one bit in the bitmap for each country where releasability is permitted. The fundamental problem with this approach concerns the correct application of a text label to the data when printed or displayed. Standard Oracle and Data General routines can translate a label into a displayable text string, but compartment names are associated to each bit in the bitmap, and assigning, for instance, 'REL UK' to the bit for releasability to the United Kingdom would create a

problem for data marked releasable to NATO. The resulting displayed string would read 'REL UK, REL FR, REL GE, REL BE, and on and on. A better approach is to map the null string to the bits for country releasability, but to set aside the final eight bits to be used by ENGRAFT software as a key into a table which lists the text for the displayed string. The text lookup itself is handled by an ENGRAFT application.

Biographies:

Mike Lefler, PRC Technology Center, PRC Inc., 1500 PRC Drive, McLean, VA 22102, 703-556-1863 (lefler_mike.@prc.com). As a Senior Technical Fellow Mr. Lefler serves as an in-house consultant on database technology for PRC, a major systems integrator. In this capacity he supports a wide spectrum of PRC programs. In addition he edits a column devoted to DBMS technology for PRC's *Technology Transfer* newsletter, and he has presented seminars on relational database technology, parallel database technology, information engineering methodology, and distributed databases as part of PRC's internal training program.

James Bradley, Coleman Research Corporation, 9891 Broken Land Parkway, Columbia, MD 21046, 410-691-5242 (JFBradley@ aol.com). Mr. Bradley is a Senior Software Engineer leading the ENGRAFT Security Engineering team.

ON-LINE ANALYTICAL PROCESSING (OLAP) AS A DATA ACCESS METHOD

Maureen K. Armacost
Richard S. Carson & Associates, Inc.

August 25, 1995

ON-LINE ANALYTICAL PROCESSING (OLAP) AS A DATA ACCESS METHOD

Maureen K. Armacost, Richard S. Carson & Associates, Inc.

Abstract

On-line Analytical Processing (OLAP) is a technology that only in recent years has become available as a tool to support data analysis. It offers some very powerful capabilities for allowing data analysts to survey information from a variety of viewpoints. No longer is information conceptualized in a series of rows and columns; rather it is viewed as an n-dimensional cube, with depth defined at each row and column. An analyst can re-order information presented simply by grabbing the headings and moving them from columns to rows, or switching them back (referred to as pivoting data). Multi-dimensional analysis "maximizes data relationships to find any and all patterns and trends." ("All About Multi-dimensional Technology," *Client/Server Today*, March 1995.)

Multi-dimensional analysis is most directly suited for financial applications. This is due to the nature of defining and populating the dimensions of data. However, the toolsets are expanding to be able to support various Boolean types of data, such as that typically found in a relational database. This will result in expanded use of the tool to more ad hoc queries and questions, providing greater flexibility to the data analyst looking for trends and patterns in data.

In providing such a capability, the data administrator is required to "undo" everything ever learned about relational databases. Multidimensional databases represent every possible value of a data element in order to build the database dimensions. This allows all the permutations of a data set to be predefined, thus providing the quick response needed for the consolidation and drill-down capabilities so important to on-line analytical processing.

In this paper, the author will present OLAP as a data access method. The discussion will include a survey of how it is being applied in the business community, including both financial and non-financial application areas. The author will discuss its potential uses in the Department of Defense (DoD) and how it is starting to be applied in some areas of the DoD. Finally, the author will discuss the unique challenges OLAP presents for data administrators in managing data with such a tool.

1. INTRODUCTION

One of the biggest problems today, facing those responsible for analyzing data, is the ability to harness the data in such a way that it can be examined and analyzed with meaningful results. Decision support tools are designed for this, but traditionally have required a significant amount of programmer intervention.

With the advent of client/server technology, and the explosion in commercial software products available in the marketplace today, one specific type of tool has surfaced that has some very powerful capabilities for analysis and data access, and benefiting those requiring decision support capabilities. That tool is called on-line analytical processing (OLAP), or multi-dimensional analysis. Using OLAP, information is viewed and maintained in an n-dimensional cube, to maximize the relationships that can be found across data sets. This permits the data analyst to identify trends, and to look for patterns in data.

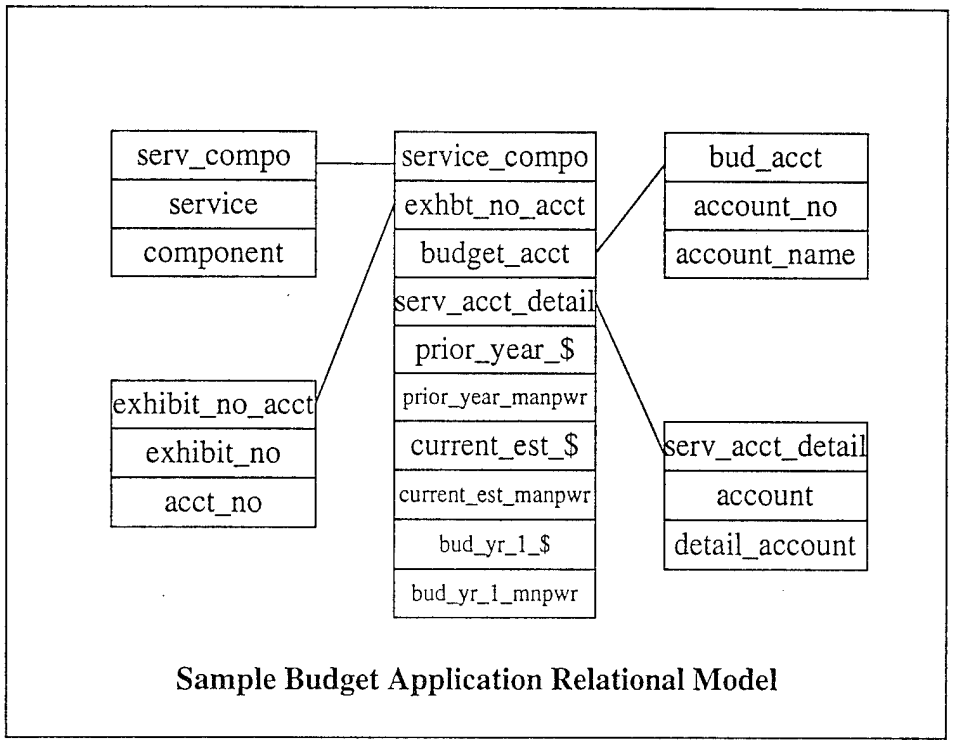
The following paragraphs provide an overview of OLAP, including a discussion of how it is currently being used in the business community, its potential application to DoD, and how it is beginning to be applied in selected areas of DoD. This paper also provides a discussion on the unique challenges OLAP presents to the data manager in using and maintaining the data administration tools.

2. OVERVIEW OF OLAP

Traditionally, data administrators think of data in terms of the relational model, in two dimensions, but allow row and column sets to have relationships to other row and column sets, such that if all row and column sets were combined, the result would be a very large table of rows and columns. With OLAP, these flat tables are transformed into cubes, where sets of rows and columns now have depth. This depth can be defined in terms of any number of attributes.

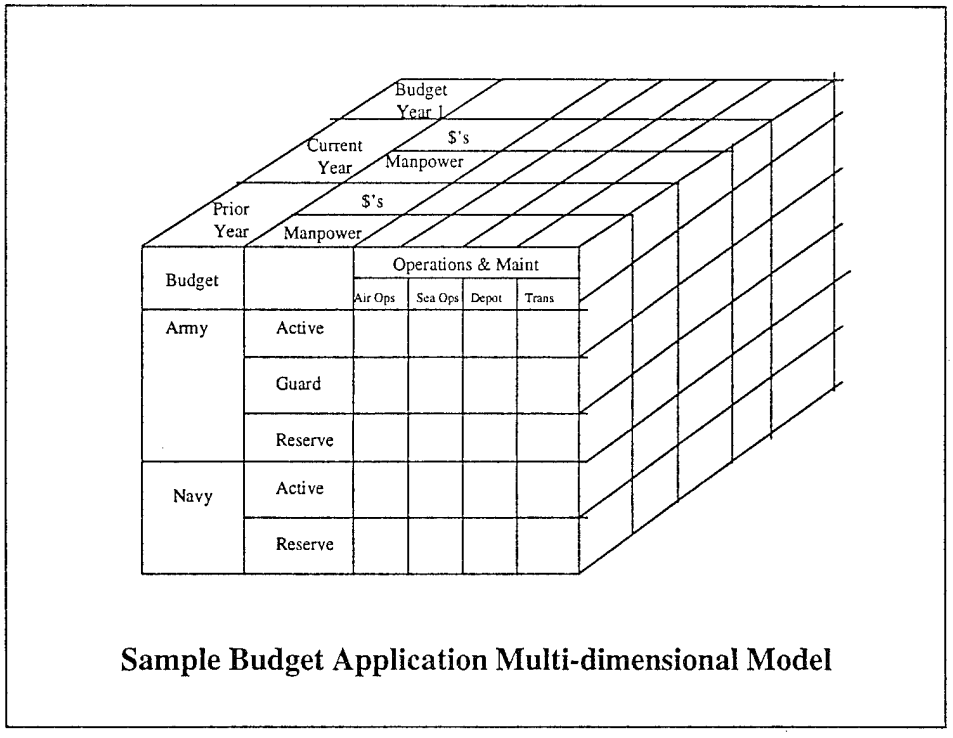
Given this added dimension to data, the tools providing an OLAP capability have been designed for instant access to any cell location within the cube. The tools also provide the ability to quickly summarize large sets of data, based on the pre-defined relationships established in the cube.

As an example, take a DoD budget application. In a relation database, tables would be defined that maintain information on services and components, budget exhibits, budget accounts, and prior year, current year and budget year manpower and dollar expenditures. This might be represented as in Figure 1. In an OLAP database, the cube for that data set my be represented as shown in Figure 2.



Sample Budget Application Relational Model

Figure 1



Sample Budget Application Multi-dimensional Model

Figure 2

With OLAP, all possible permutation of data relationships are identified and captures ion the database, so that the data may be more quickly retrieved and analyzed. Most OLAP tools also provide

the ability to define formulas. As a result, using the example in Figure 2, a budget expansion can be defined as the difference between budget year 1 and current estimate. Most OLAP tools provide the flexibility to generate formulas based ion standard mathematical functions, such as add, subtract, multiply and divide. This provides the data analyst the added capability to quickly identify and analyze trends associated with variances and other functions.

When the cube is populated, data is captured at each unique intersection point in the cube. As a result, some locations in the cube may be sparsely populated, while others are heavily populated. Most OLAP database tools are optimized to maintain this type of data and are able to easily manage dense versus sparse populations of data in the cube.

3. CURRENT USES OF OLAP

OLAP was originally targeted for the financial community. Its rapid retrieval and data consolidation capabilities make it directly suited towards the needs for financial analysts. Those capabilities include consolidating numbers and showing variances. There are a number of Fortune 500 firms using OLAP to assist in analyzing financial data. It is being used to analyze and monitor sales information, corporate financial conditions, budgets and budget variances, receivables, etc. The tools available for frequent data refresh mechanisms to support detailed analysis efforts.

OLAP is not designed to replace traditional corporate data collection mechanisms, such as transaction databases, rather it is designed to augment the suite of data access and data analysis tools available to those that perform such tasks. Typically, corporations using OLAP maintain transaction-based information in a traditional relational transact database. OLAP-based products provide tools that allow the data to be consolidated to a next higher level, and the mapped into a multi-dimensional database.

From the data analyst perspective, data is presented using tools and techniques with which one is already familiar. Many tools use a spreadsheet type of interface with added capabilities unique to OLAP. These added capabilities include:

- the ability to pivot data, including swapping rows and columns if data, so it may be viewed from a different perspective;
- the ability to drill down, expanding the detail provided in rows and columns (adding that next dimensions); and
- the ability to identify when no data is available from a selected intersection point in the cube.

4. APPLICABILITY TO DOD

OLAP was direct applicability to those organizations responsible for analyzing larger sets of data for trends and aberrations, including financial organizations (such as the comptroller), policy-making bodies, and staff organizations. DoD is just beginning to experiment with it in two areas: budgets analysis and readiness analysis.

In the area of budget analysis, OLAP databases are being established from source data external to the organization, which is provided for policy-making and program analysis purposes, The budget data provided is captured over time and include:

- fiscal

- program element code (which can be further dissected
- appropriation
- budgeted dollars
- actual dollars

Using OLAP, the analyst has the ability to quickly identify trends over time for budgeted and actual dollars by appropriation and program element code. Analysts are able to use their standard spreadsheet interface for analyzing data, and also have the ability to quickly pivot data and to drill down for more details.

The advantage of using OLAP is that is does not require programmer intervention. Once the multi-dimensional database has been defined, and the appropriate software interfaces are configured at the user desktop, no programming is required. Yet the interface tools are powerful enough to allow users to pivot data, drill down for more details, chart the results of analyses, and develop standard reports.

In the area of readiness, OLAP is beginning to be used to examine historical trends and to analyze current readiness information. Readiness data is captured at a very detailed level, so it may be easily used in a multi-dimensional database. Using OLAP, the readiness information can be used by a policy-making organization to examine trends and aberrations. Readiness information is reported at a significant level of detail, allowing for flexible pivoting capabilities using OLAP interface tools. This provides the data analyst powerful tools for examining readiness information.

OLAP is also currently being examined to determine how it may be applied to projection functions, specifically in projecting readiness impacts to planned future resources. Using the n-dimensional cube, the relationships between parameters can be defined and populated with the supporting data over a future period of time. Using the OLAP analytical tools, one could then identify changes to parameters (such as changes to the budget), and then model the results.

5. CHALLENGES TO ADMINISTRATORS

For the OLAP database administrator, learning to create a multi-dimensional database requires "unlearning" everything ever learned about relational database design. Relational databases are of two dimensions: rows and columns. Multi-dimensional databases can be thought of in terms of a cube, adding depth to the rows and columns.

For many, this may be a difficult transition, primarily because it requires making assumptions about the data contents. Relational database design has taught us to remove those assumptions with regard to content from the database. Fortunately, the OLAP tools available in the marketplace today, allow retrieval from traditional relational databases using SQL, so the process can be simplified by mapping SQL queries to specific components of the multi-dimensional database.

The tools available in the marketplace that support OLAP have several criteria that distinguish them:

- single-user vs. multi-user
- client-based vs. server-based
- level of security required

With a large user community, sharing a large data set where security and data protection is required, a server-based multi-dimensional database may be appropriate. In a situation where individual users are

working with stand-alone, small data sets, it may be more appropriate to use single-user, client-based tools. As with most situations, tool selection is dictated by the requirements.

6. THE USER PERSPECTIVE

Users working with OLAP tools see a familiar interface. Data is typically presented in rows and columns. Drilling down to greater levels of detail is accomplished by selecting a dimension (or attribute) and expanding it. Most tools provide features to develop graphical representations of data displayed. For example, Figure 3 represents the budget application described above, as the user might view it. Figure 4 depicts how the data might be represented when the user drills down to view details associated with the services. Figure 5 represents what the user might see were the data to be pivoted. Note the services have shifted from rows to columns and the years have moved to rows.

	Prior Year	Current Year	Budget Year 1
	O&M	O&M	O&M
All Service	100	125	150

Figure 3

			Prior Year	Current Year	Budget Year 1
			O&M	O&M	O&M
All Service			100	125	150
	Army		50	65	75
		Active	30	40	45
		Guard	10	12.5	15
		Reserve	10	12.5	15
	Navy		50	60	75
		Active	30	40	50
		Reserve	20	20	25

Figure 4

		Army			Navy	
		Active	Guard	Reserve	Active	Reserve
O&M						
	Prior Year	30	10	10	30	20
	Current Year	40	12.5	12.5	40	20
	Budget Year 1	45	15	15	50	25

Figure 5

7. CONCLUSION

OLAP provides an effective tool for an alternative decision support capability. It is powerful, and the interface tools are based on standard decision support capabilities already in the marketplace. It requires data administrators to think in new ways about their data, however the effectiveness of the tools provide a significant benefit for the user. OLAP provides a powerful capability that is starting to be used to support many policy-making and financially-based decision support efforts.

Biography

Maureen Armacost is a Senior Associate at Richard S. Carson & Associates, Inc., where she has served as Task Leader and Project Manager for the past seven years. A recent project of hers was recognized as one of *INFOWorld*'s top ten client/server initiatives. Ms. Armacost has over twelve years experience in life cycle development and database management systems development, with applications experience in the intelligence and defense communities. A graduate of Boston College, she is currently pursing a Master's Degree at Johns Hopkins University.

Maureen K. Armacost
Senior Associate
Richard S. Carson & Associates, Inc.
4330 East-West Highway, Suite 304
Bethesda, Maryland 20814

301.656.4565 x209
301.656.4806 (fax)
armacost@carsoninc.com

LESSONS LEARNED IN LEGACY DATA ACCESS

by
Diane C. Reilly, Richard S. Carson & Associates, Inc.
Jeanine A. Fleming, Richard S. Carson & Associates, Inc.

ABSTRACT

Over a billion lines of code, inconsistent data, lack of data standardization, old technology, and poor documentation are key characteristics of legacy systems. As the Department of Defense (DoD) makes strides to integrate stovepipe systems and standardize data, functionals as well as users of data are still faced with the problems underlying legacy systems. Because not all of the information is accessible in one place, development of several queries is necessary to access multiple platforms, and the results require manual integration. Hence, questions requiring more than one database are put in the "too hard" pile.

New technology can provide access to multiple legacy databases across multiple platforms, but simply building the link won't necessarily fix the problems with legacy systems. Users may be provided with a transparent link that results in invalid data because different legacy systems use different naming conventions and different types and sizes.

To resolve these problems, the user must have an in-depth understanding of the legacy databases and the environment in which they actually reside. With this understanding, various crosswalk tables, validation routines, and other tools can be developed to quickly identify inconsistencies and ensure that the user receives valid data.

Legacy systems need to be integrated, migrated, and re-engineered to standardized data before it is made available across DoD. Until this is achieved, users need an interim solution that allows a way to integrate and access legacy systems using "live" data in real time to achieve needed results. In this paper, the authors address a general background of the problems underlying legacy systems, an approach to integrating multiple legacy system databases, a discussion of tools used for integration and access of legacy data, and lessons learned from integrating over twenty-five legacy system databases for the Department of the Army, Deputy Chief of Staff for Operations and Plans (DCSOPS).

LESSONS LEARNED IN LEGACY DATA ACCESS

by
Diane C. Reilly, Richard S. Carson & Associates, Inc.
Jeanine A. Fleming, Richard S. Carson & Associates, Inc.

1. Background

Over a billion lines of code, inconsistent data, lack of data standardization, old technology, and poor documentation are key characteristics of legacy systems. As the Department of Defense (DoD) makes strides to integrate stovepipe systems and standardize data, functionals as well as users of data are still faced with the problems underlying legacy systems. Problems such as different naming conventions, different types and sizes, and erroneous data due to the lack of validation and verification when data is entered. In addition, legacy systems reside on multiple platforms such as mainframes, mini-computers and PCs, which results in different database formats that are not compatible to one another. Because the information is not accessible in one place, the development of several queries is necessary to access multiple platforms, and the results require manual integration. Hence, questions requiring more than one database are put in the "too hard" pile.

The Department of the Army, Deputy Chief of Staff for Operations and Plans (DCSOPS) needed a way to integrate multiple legacy system databases in order to rapidly respond to those questions that were normally put in the "too hard," pile, questions which were in response to highly visible studies within the Department of the Army and Congressional queries. Since most of the databases resided on different mainframes across the country with no direct access to the data in most cases, it was very difficult for users to respond to management requirements in a timely manner. Most of the query results from each of the systems were integrated manually and re-typed into other analysis tools for further manipulation.

Integrating "snapshots" of these databases in to a central repository provided the users direct access to the data in real time, greatly reducing the amount of time spent in manually integrating the data and allowing results to be presented in a way that supported upper management strategic decisions.

2. Approach to Integrating Data

When we at Carson Associates began this integration project in 1987, the task at hand was to develop an interim but flexible solution until the Army legacy system databases were integrated, migrated, interconnected, and the data standardized. This task required integrating over twenty-five Army-wide legacy system databases to include personnel, transportation, logistics, force structure, and transportation with approximately 300 tables and over 4,000 data elements. This classified system, recently migrated to client/server using RISC machines and 486 compatibles, was comprised of eight workstations, a communications server, a fileserver, a database server, and various printers.

This particular office within the DCSOPS was not the "creator" of the data but a "user" of the data. The first obstacle we had to overcome was obtaining a copy of the data from each of the data proponents. We learned that the thought of sharing data is a hard concept to swallow, even still today, for fear someone will route around in the data and find something out. We also learned that there were multiple personnel and equipment databases and had to determine which ones the Army used as the authoritative source, and that depended on the person asked on any given day.

Data available to us was not in a standard format, but in a variety of formats to include spreadsheets, EBCDIC files, ASCII files, dBase files is varied media, such as 9-track tapes, 8mm and 150 MB tapes, floppies, Bernoulli, and recently, CD-ROM. This became an issue because at the time we did not have the capability to support all media. Another problem we faced is the lack of documentation associated with each of the legacy systems. At most, we received the file format in order to transfer the data on to our system. In some cases, when translating the data to the system, we had to develop specialized data translation routines to split out data when more than one record type was in a given file, or the data was not in fixed format. As we built the database, we did not formally integrate and normalize the data in the traditional sense to reduce redundancies. However, we integrated the data in its "natural" state into one large database and partitioned the database into categories to represent each of the different legacy systems. By implementing one large database, we were able to easily relate the databases and increase performance of the database.

At first glance, we noticed that the data elements were not standardized and data elements that were related to one another were of different types and formats with different naming conventions. For example, a unit was identified as a UIC, unit, or unit_id. To resolve the problems of different types and formats, we created crosswalk tables that would allow the different databases to relate to one another. The different naming conventions were not of any consequence because as long as the data was the same in terms of content, type and format, we could relate the data. We built an on-line data dictionary that documented the relationship between the different data elements and their respective tables.

The DCSOPS purchased a proprietary tool called the IBM/Metaphor Data Interpretation System (DIS), and used it to access the integrated data. DIS is a front-end graphical user interface (GUI) that gives the user a suite of tools to access and manipulate data. At the time of this integration project, DIS was the only tool that provided the user the capability to query the database and download the results to a spreadsheet or graphic without any programming or manual integration. Since then, advancements in technology have provided many other tools that can be integrated, and can provide the user with similar capability.

There are other approaches to giving users direct access to data. In particular, integrating a suite of commercial off-the-shelf (COTS) tools and middleware can provide the user a user-friendly interface that makes data integration transparent to the user.

3. Data Access

A significant problem with legacy databases is that they most likely reside on multiple platforms. Many of the older databases still reside on a mainframe. Many newer databases are located on PCs in simple spreadsheet files, such as LOTUS, Microsoft Excel, or dBase. The task is not as simple as providing a simple front-end graphical user interface; rather, the user must have access across these various platforms, and this connectivity should be transparent to the user. A variety of commercial-off-the-shelf software tools are available to provide faster access to multiple legacy databases. Some tools that provide this type of access include, but are certainly not limited to, Microsoft Access, Powersoft Corporation PowerBuilder, and the development tool, Visual Basic. These tools have one common characteristic: they all provide a transparent user interface. With one click of a button, the data analyst can view extracts of data from multiple databases. What actually happens behind the scenes is that an access tool connects to the legacy databases, retrieves the user constrained data, and downloads the data to a spreadsheet, document, or graphic for further manipulation.

This means no more long waits to receive printouts from independent platforms and manually integrated data. The manual integration increases the probability of errors in data, while electronic integration reduces the probability. However, there exists a common misunderstanding that faster is better. In cases where there are underlying problems, "faster" just means generating erroneous results all that more rapidly. In the end all that these tools are doing is laying new technology over existing problems, thus masking the problems and increasing the probability of generating additional problems. The user can now see the problems within the data more quickly.

Therefore, it is not enough to provide a user with faster access to the various legacy databases required to perform his/her job. The user also needs to have access to valid data. The various problems currently existing within the legacy databases do not simply disappear when a user can access them seemingly instantaneously. In order to provide solutions more quickly, Carson Associates has developed a data validation approach to help ensure that, until legacy databases can be totally re-engineered, end-users will receive valid data. Our data validation approach consists of the following steps:

- Review system documentation and system environment
- Interview technical and functional experts
- Develop a data dictionary showing relationship between disparate databases
- Develop validation and verification routines and test for validity of data
- Create crosswalk table to relate disparate tables
- Report data inconsistencies to "owners" of data

One of the lessons learned by Carson Associates is that data validation routines should be developed to identify inconsistencies in the data. These inconsistencies can be caused by processing errors, data entry errors, or different date stamps. The validation routines have to be developed with knowledge of the legacy databases and the environment in which they reside; otherwise they might generate more problems than they resolve by identifying errors that do not

actually exist. For example, comparing data from two different legacy databases with dissimilar results does not mean, necessarily, that one of the tables is corrupt. It might just mean that the time frames for which the data is valid in the tables are not the same.

The best place to start learning about the environment of a legacy database is to study any documentation available for the database. Important sections that might be available in the documentation include the following:

- a data dictionary describing data elements
- a description of inputs to the database, such as other databases, user input, etc.
- a description of outputs to other databases or simply an electronic format
- a description of what the database was designed to achieve
- a valid time frame for the data, i.e. data stamp

In many cases, however, the documentation for a legacy database is incomplete because it has not been required or needed. Additional sources for information regarding a database are the technical and functional experts who maintain the database or the system generating the database.

The major key to understanding each legacy database is to research the data elements. This includes more than determining the data type and length of the element. The developer of the validation routine should know, when applicable, the valid values for the data element. For example, the United States has four military services, the Army, Air Force, Navy, and Marine Corps. If the database had a data element identified as service, the developer should determine if this field only includes the United States four services or if the database might also include pseudo-services, in times of war, such as the US Coast Guard and Merchant Marines. A complete data dictionary should be developed, if one does not already exist. If a data dictionary does already exist, it most likely will be incomplete. The dictionary should identify what each data element represents and fully describe the element, i.e. data type and length. This is important because other legacy databases might identify the same data element by a different name. In order to join the disparate tables, it is important to find these similarities. If more than one table exists within the database, the table(s) in which the element resides should also be identified as well as the type of data grouped within the tables.

By understanding the environment of each legacy database, we can more easily identify what types of validation routines should be developed. For those data fields with discrete values, the data field for each record can be checked to determine if its value is valid. In order to maintain the integrity of the original legacy database, the tables in the original database are not modified should a data field entry be found to be incorrect. Rather a lookup table is created that contains the new valid values by record, as well as the original value. This table then serves as a reference report for those individuals responsible for the system generating the database. The invalid data can be evaluated and modifications to the environment of the legacy database can be made. Based on the results from a validation routine, we found that a particular Army force structure database had an error in its endstrength of over 50,000. After further analysis, it was determined that a data entry error was made. This was reported to the data proponents, and the database was modified to reflect the correct value.

Additional validation routines can be developed to verify summation values, provided the known values are available. While this is obviously a check on the integrity of the legacy database from which the summations are derived, it can also provide a baseline for other legacy databases related to the summation database. This means that if another legacy database also contains the field(s) by which the summed values were obtained, totals can be derived and compared between the related databases. It might not be necessary for these summed values from the separate databases to be equal because we have found that related but disparate databases do not usually account for data in the same manner. Another reason that data totals might not match is that the date stamps are different. It would be illogical to assume that data from different time frames would be equal. If any documentation exists for the databases, it should be reviewed to provide help in these areas.

In order to join data from these disparate tables, the database analyst must look at the data elements in all tables very closely. While two tables may have data elements named exactly the same, in many cases the data type and/or data size are different. To resolve these issues crosswalk tables can be developed to include the disparate data elements. For example, several of the Army databases have a field "SRC" that is defined as a structure requirement code. The SRC code may be 9, 11, or 13 characters in length depending on the database. In order for the different databases to be linked or related, a crosswalk table was developed with all three lengths of the SRC. The crosswalk table can then be used to extract the information from the separate tables.

Legacy databases usually have many lookup or referential tables that are used to generate a single large flat table. These smaller tables provide an additional resource for understanding the legacy databases. Another lesson learned by Carson Associates was to develop small relational tables that help speed access to the data, depending on the nature of the queries to be generated. Examples include splitting data by fiscal year, by Component code, or by any other identifiable record type, i.e., if the legacy database has a data element specifying the data type.

4. Conclusion

In conclusion, don't assume laying new technology over legacy systems will resolve the underlying problems of legacy systems. Faster data access is good but not the final solution. Our approach may not resolve all the underlying problems of legacy systems but it can ensure the probability of users receiving invalid data is reduced. To date, our approach has worked successfully in the Department of the Army.

BIOGRAPHY

Diane C. Reilly holds a Bachelor's of Science Degree in Management Science, from the University of Maryland, where she specialized in Decision Information Sciences. Currently with Richard S. Carson & Associates, Inc., a management consulting firm, Ms. Reilly has over ten years experience including business process and data re-engineering, data integration, data migration, systems migration, systems software development using client/server technology, and management.

Diane C. Reilly
Senior Associate
Richard S. Carson & Associates, Inc.
4330 East-West Highway, Suite 304
Bethesda, Maryland 20814

(301) 656-4565
fax: (301) 656-4806
reilly@carsoninc.com

A graduate of The George Washington University with a Bachelor's of Science Degree in Systems Analysis and Engineering (Operations Research), Jeanine A. Fleming is currently pursuing a Master's Degree in Operations Research with emphasis in Management Sciences. Ms. Fleming has over five years of experience with Richard S. Carson & Associates, Inc., in systems software development and data analysis in a client/server environment.

Jeanine A. Fleming
Associate
Richard S. Carson & Associates, Inc.
4330 East-West Highway, Suite 304
Bethesda, Maryland 20814

(301) 656-4565
fax: (301) 656-4806
fleming@carsoninc.com

THE IMPACT OF WORKFLOW ON DATA MANAGEMENT

D. D. MARKS AND J. K. WASS, RICHARD S. CARSON & ASSOCIATES

1. INTRODUCTION

The emergence of Database Management Systems (DBMS) in the 1970s and 1980s removed the management of data from software applications. The rules of data management became standardized in the DBMS and not redundantly embedded in modules of application code.

A comparable revolution may be underway due to the introduction of Workflow systems. These emerging technologies focus on the coordination of tasks and the transfer of control thereby permitting us to separate the management of processes from application modules.

Data management benefited significantly from DBMS introduction. Is Workflow a data management ally as well, or does it pose a threat? Both the role of Workflow in data management and Workflow's impact on the management and administration of information resources are uncharted territory. We will examine the impact of new Workflow technologies on data administration and application interconnectivity.

We will test the use of the Zachman Information System Architecture (ISA) framework in describing the impact of these new technologies. We will evaluate whether additional views are required to make the Zachman model inclusive and coherent with respect to the new technology of Workflow.

After validating the architecture we will account for the paradigm shift that workflow brings to information management and enterprise integration and reveal approaches for implementation of Workflow that advance Data Management.

2. THE ZACHMAN ISA FRAMEWORK

The Zachman Information System Architecture (ISA) Framework was developed as a model for information system planning. J. F. Sowa and & J. A. Zachman describe it as a tool for "segmenting the descriptions of the enterprise: for separating independent variables into understandable, designable components; for developing appropriate design formalisms; and for establishing an enterprise infrastructure in which change can be assimilated in a manageable fashion." ["Extending and formalizing the framework for information systems and architecture," *IBM Systems Journal*, Vol 31, No 3, 1992.] Sowa and Zachman detail five modeling views and six enterprise model types used to design and specify an information system. Figure 1 is derived from the Zachman "generic" Information Systems Architecture (ISA).

Zachman ISA Views

The ISA includes the scope, enterprise, system, technology, and component views which progress in detail from system conception to implementation. The Scope is an executive summary for a planner to estimate what a system might cost or how it would perform. The Enterprise Model View constitutes the owner's view of the business and shows the interaction of the business entities and processes. The System Model View (or Designer's perspective) is created by the systems analyst who must determine what data elements and functions derive from the business entities and processes. The Technology Model View (or Builder's perspective) adapts the system model to the constraints of the programming languages, hardware, etc. The Component Model View (or the Sub-Contractor's perspective) holds the detailed specifications given to the programmers who will code individual modules.

	DATA *Entity* *Relationship*	**FUNCTION** *Function* *Argument*	**NETWORK** *Node* *Link*	**PEOPLE** *Agent* *Work*	**TIME** *Time* *Cycle*	**MOTIVATION** *Ends* *Means*
SCOPE Planner	LIST OF THINGS IMPORTANT TO THE BUSINESS *E: Class of business thing*	LIST OF BUSINESS PROCESSES *F: Class of Business Process*	LIST OF LOCATIONS *N: Major business location*	LIST OF AGENTS/ ORGS IMPORTANT TO THE BUSINESS *A: Major Organization Unit*	LIST OF EVENTS SIGNIFICANT TO THE BUSINESS *T: Major Business Event*	LIST OF BUSINESS GOALS/ STRATEGIES *E/M: Major Bus. Goal/Critical Success Factor*
ENTERPRISE MODEL Owner	E/R DIAGRAM *E: Business Entity* *R: Business Constraint*	PROCESS FLOW DIAGRAM *F: Business Process* *A: Business Resources*	LOGISTICS NETWORK *N: Business Location* *L: Business Linkage*	ORGANIZATION CHART *A: Organization Unit* *W: Work Product*	MASTER SCHEDULE *T: Business Event* *C: Business Cycle*	BUSINESS PLAN *E: Business Objective* *M: Business Strategy*
SYSTEM MODEL Designer	DATA MODEL *E: Data Entity* *R: Data Relationship*	DATA FLOW DIAGRAM *F: Application View* *A: User View*	DISTRIBUTED SYSTEM ARCHITECTURE *N: I/S Function* *L: Line Characteristics*	HUMAN INTERFACE ARCHITECTURE *A: Role* *W: Deliverable*	PROCESSING STRUCTURE *T: System Event* *C: Processing Cycle*	KNOWLEDGE ARCHITECTURE *E: Criterion* *M: Option*
TECHNOLOGY MODEL Builder	DATA DESIGN *E: Segment/ Row* *R: Pointer/Key*	STRUCTURE CHART *F: Computer Function* *A: Screen/ Device Fmt*	SYSTEM ARCHITECTURE *N: Hardware/ System Software* *L: Line Specifications*	HUMAN/ TECHNOLOGY INTERFACE *A: User* *W: Job*	CONTROL STRUCTURE *T: Execute* *C: Component Cycle*	KNOWLEDGE DESIGN *E: Condition* *M: Action*
COMPONENTS Subcontractor	DATA DEFINITION DESCRIPTION *E: Field* *R: Address*	PROGRAM F: Language Stmt A: Control Block	NETWORK ARCHITECTURE *N: Address* *L: Protocol*	SECURITY ARCHITECTURE *A: Identity* *W: Transaction*	TIMING DEFINITION *T: Interrupt* *C: Machine Cycle*	KNOWLEDGE DEFINITION *E: Subcondition* *M: Step*
FUNCTIONING SYSTEM	*DATA*	*FUNCTION*	*NETWORK*	*ORGANIZATION*	*SCHEDULE*	*STRATEGY*

Figure 1 - Zachman Information Systems Architecture Framework

Zachman Model Types

Each of the five ISA views is cast against six columnar questions which elicits a model type. "What?" deals with data modeling. "How?" is function and process modeling. "Where?"

describes physical locations ranging from street addresses down to network architectures. "Who?" looks at people who perform work. "When?" asks about time both in execution times and sequences and in durations. "Why?" denotes the motivations unique to each view.

Zachman ISA Rules

The Zachman matrix is governed by seven rules (which are italicized):

Rule 1 *The columns have no order.* (Though the rows progress downward in implementation detail.)
Rule 2 *Each column has a simple, basic model.* The six models currently accounted for by Zachman are Data, Function, Network, People, Time, and Motivation.
Rule 3 *The basic model of each column must be unique.* Neither the name nor the concept underlying any entity or model connector is repeated in the matrix.
Rule 4 *Each row represents a distinct, unique perspective.* Each perspective reflects a different set of constraints.
Rule 5 *Each cell is unique.*
Rule 6 *The composite or integration of all cell models in one row constitutes a complete model from the perspective of that row.*
Rule 7 *The logic is recursive.* (This suggests that ISA can represent any information system's "enterprise" at any level and can describe any type of automated implementation.

Is the Zachman ISA Framework rich enough and flexible enough to accommodate workflow? Each row represents a particular view of reality; will the reality of workflow fit into the existing model? Do any peculiar design and specification needs govern the implementation of workflow? If a workflow-specific view is inserted into the Zachman matrix will the rules be honored? Will existing views separate cleanly or will we tear the fabric?

3. WORKFLOW

Workflow Defined

Enterprise Coordination Logic. Workflow provides the coordination logic of the enterprise that enables and guides functionals as they conduct business.

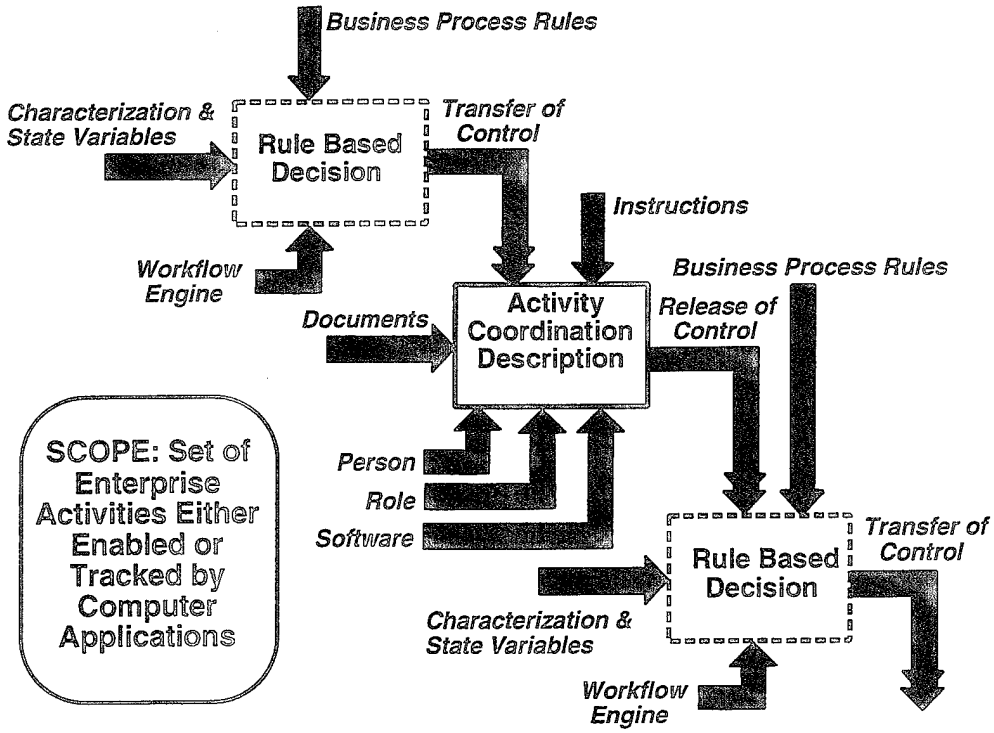


Figure 2 - A Workflow Perspective

Transfer of Control. *Work* happens at the "desktop" (or the shop floor). Workflow is the transfer of control of tasks between and among "desktops". Workflow is not about work itself but rather about the sequence of execution of units of work. Figure 2 is an "IDEF-like" model with a workflow perspective. The double-headed arrows represent transfer of control between

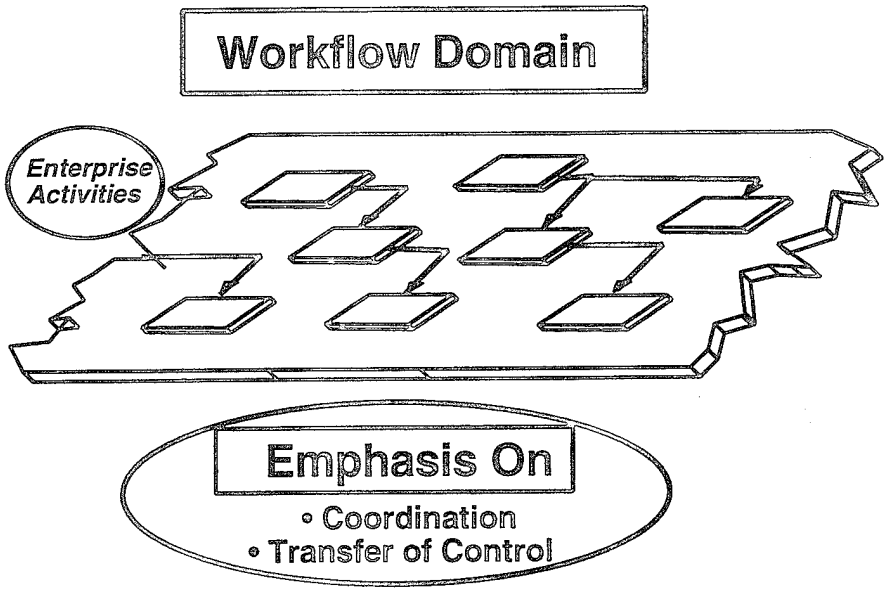


Figure 3 - An Architecture for Flow Support

activities. Figure 3 describes an enterprise architecture for support of workflow. Various sequential and parallel business activities exist throughout the enterprise. They are linked by transfers of control which can range from paper folders or route slips to highly automated workflow systems.

The Purpose of Workflow

Coordination. Workflow is about coordination of inputs, controls, constraints, and resources so an activity may produce an appropriate output or deliverable. Or in the language of the DoD Enterprise Model: Coordination enables a capability.

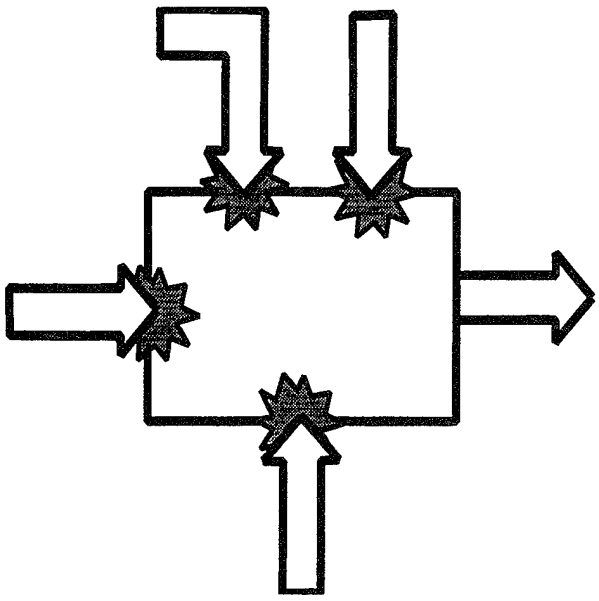


Figure 4 - Coordination: Enabling Capability

Capability. Capability is "the potential ability to do work, perform a function, achieve an objective, or provide a service of importance to the DoD Mission." In short capability is everything needed to produce an output. The workflow engine assembles all inputs, controls, and mechanisms so that an output can be produced. If at least one critical dimension of the model's instance is missing, then work does not proceed. Coordination is most noticeable by its absence. When the purposeful relationships between inputs, controls, and mechanisms of an activity are violated some combination of quality, schedule, or cost is negatively impacted.

Workflow Architecture

Figure 5 is a typical configuration of workflow components. Workflow must account for each participant desktop and the interconnectivity between and among those desktops as well as the management of characterization and state data. The workflow engine interprets process definitions, routes work, and invokes application software according to roles and workflow control data.

The user interaction with Workflow is described in Figure 6. The Workflow user or participant initially views the process through a work list (often called an "in-basket") from which a process instance is selected or accepted. The work list contains tasks which the user is eligible to perform based on assigned role(s). The workflow system either presents documents and data directly through integrated functionality or makes them available through a launched software application. Thomas Koulopoulos [*The Workflow Imperative*, Van Nostrand Reinhold, New York (1995)] calls the document in workflow "the unifying force for all data and applications used to process data."

The user performs manual or automated tasks and may enter/edit data and create or edit

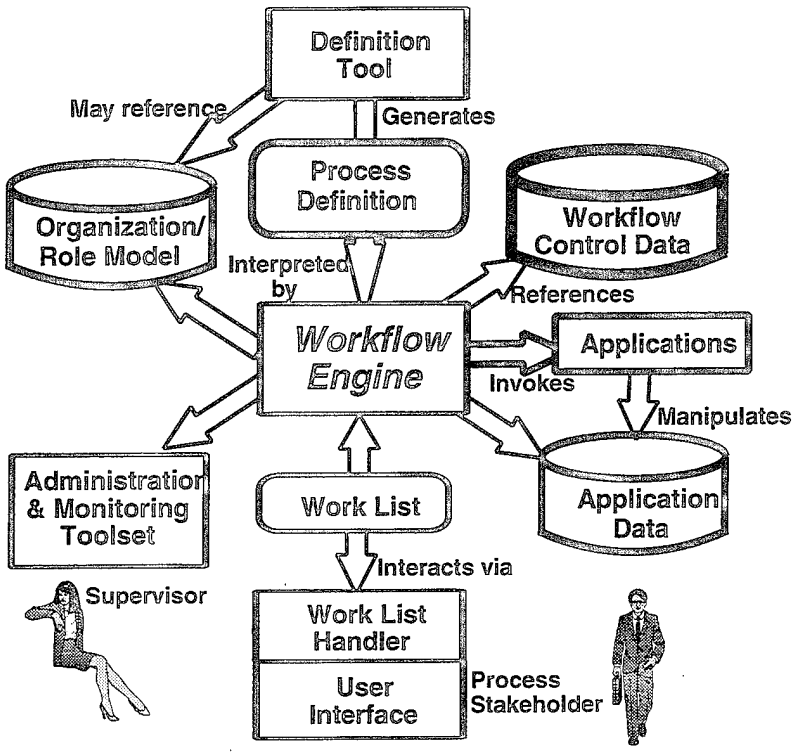


Figure 5 - Configuration of Workflow Components

documents as appropriate. When the user completes a task, the workflow engine returns the user to the in-basket to consult the task list and transfers control in the sequential process to the next desktop in the workflow.

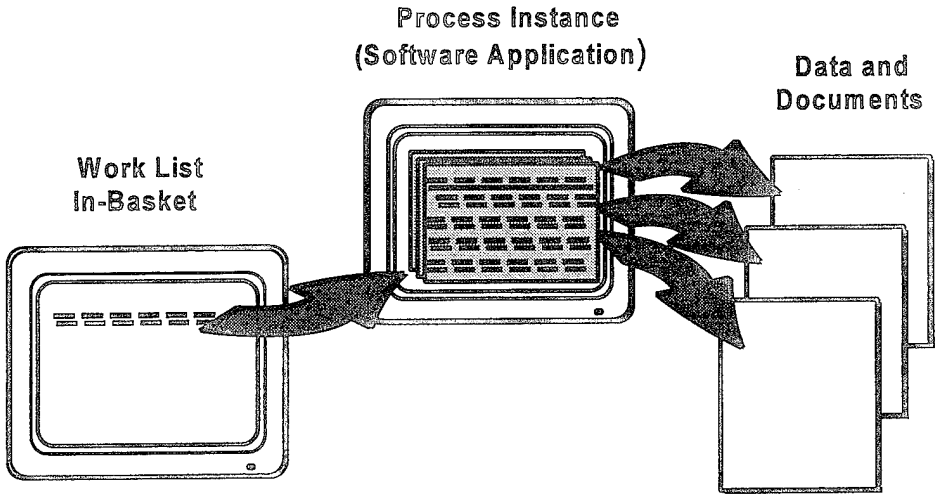


Figure 6 - User Interaction with Workflow

Workflow and ISA

To adequately test whether Zachman or any other architecture framework supports workflow implementation it is important to review what objects are needed to describe a workflow. As workflow is focused on the document and the desktop we chose to orient this schema toward the viewpoint of the workflow participant. The objects we test are:

Case Documents. Workflow originated in image management and the document focus continues in most implementations.

Task. A task is a unit of work. This represents both a low-level business function and the workflow automated implementation of that function.

In-Basket. This workflow metaphor represents the automated task list and/or menu presented to the participant.

Role/Category. Role and category describe what a user is qualified or authorized to do. The supervisor/administrator and workflow engine route work based on roles and categories.

Case Variables. These include characterization variables which describe the case and state variables which report and control pathing.

If workflow is beyond the explanatory power of ISA, then constructing and inserting a new row should be possible. If the new row (Workflow Participant View) is not unique, then one or more ISA rules will be violated. By applying ISA rule 7, the recursion principle, it could be argued that the owner view is sufficient to represent the participant. Does this argument hold? Figure 7 compares the generic ISA Owner view with a Workflow Participant view. We discovered that

	DATA *Entity* *Relationship*	**FUNCTION** *Function* *Argument*	**NETWORK** *Node* *Link*	**PEOPLE** *Agent* *Work*	**TIME** *Time* *Cycle*	**MOTIVATION** *Ends* *Means*
ENTERPRISE MODEL Owner	E/R DIAGRAM *E: Business Entity* *R: Business Constraint*	PROCESS FLOW DIAGRAM *F: Business Process* *A: Business Resources*	LOGISTICS NETWORK *N: Business Location* *L: Business Linkage*	ORGANIZATION CHART *A: Organization Unit* *W: Work Product*	MASTER SCHEDULE *T: Business Event* *C: Business Cycle*	BUSINESS PLAN *E: Business Objective* *M: Business Strategy*
WORKFLOW MODEL Participant	DOCUMENTS *E: Case Document* *R: Document Registration*	WORK FLOW MODEL *F: Task* *A: Case Variables*	FLOW *N: In-Basket* *L: Transfer of Control*	IN-BASKET *A: Role & Category* *W: Task List*	QUEUE *T: Acceptance Time* *C: Time on Task*	INSTRUCTIONS *E: Instructions* *M: Performance Criteria*

Figure 7- Comparison of Owner and Participant Views

when these workflow concepts are mapped into the Participant view they appear to address the same or similar concepts as the Owner view cells immediately above. To add a Participant view would also break Zachman's ISA Rule 4 which holds that "each row represents a distinct, unique

perspective" and Rule 5 which requires that "each cell is unique." Therefore, the addition of a Participant view is not a valid extension of Zachman, but rather an instance of the Owner view.

We tested the ISA Planner, Designer, Builder and Subcontractor views and found them to satisfactorily address workflow implementation. Space limitations restrict our ability to document these tests and they are left as an exercise for the reader.

If the Zachman ISA Framework is adequate, if a workflow implementation is just another information system architecture, what is the real impact on data management?

4. WORKFLOW AND DATA MANAGEMENT

Views of Enterprise

Figure 3 depicts our workflow domain of enterprise activities resting in a horizontal plane. Figure 8, an Architecture for Knowledge Support, represents the application domain in a vertical plane of program or menu hierarchy. Figure 9 is an integrated Enterprise Work Architecture with the workflow and application domains meeting at the point of application launches.

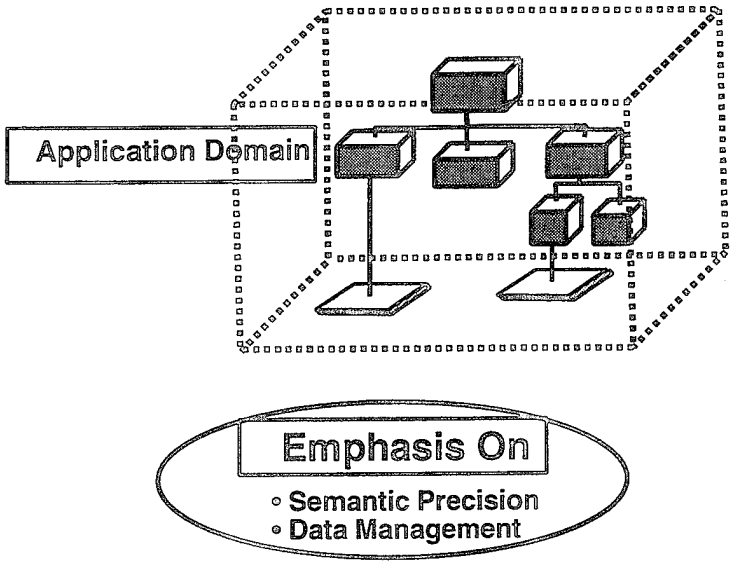


Figure 8- Architecture for Knowledge Support

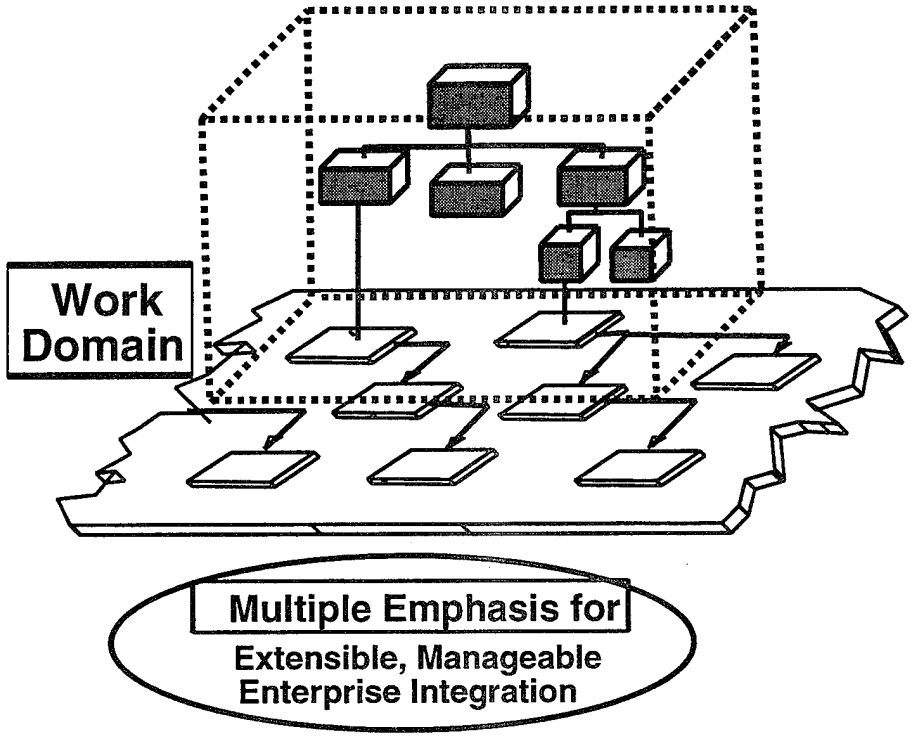


Figure 9 - Architecture for Enterprise Integration and Work Support

Data Requirements of Workflow

As shown in several of the figures, workflow invokes applications which manipulate application data. The workflow engine has its own data requirements. These data either characterize the workflow instance or describe/control the state of the workflow.

Workflow Characterization Variables. Workflow characterization variables are instances of entities and attributes typically addressed in the enterprise data model. Name of insured and claim reference numbers are examples of characterization variables in an insurance claims workflow.

Workflow State Variables. State variables are control variables that are not ordinarily included in the enterprise data model since an activity is not generally considered to be an entity. Where activities are modeled in data, the modeling is not very robust and does not usually result in part of a database structure. Where data models include business rules, these rules are typically limited to validity checks that will apply to database fields. The definition of business rules needs to be expanded to include business process rules which allow us to use state variables to determine pathing in a business process.

5. CONCLUSIONS

Workflow & Zachman.

The postulated Participant view represented the new technology of workflow as well as greater user involvement in information system specification. We had intended to prove a weakness in the Zachman ISA Framework by creating a separate model view. We found instead that the Participant view is an instance of the Owner view. We conclude that ISA is sufficiently robust to accommodate both new technology and more open approaches to system development.

Workflow & Data Management.

Workflow is not a threat to data administration. It is rather an ally and liberator. An enterprise may be integrated through process and data integration. Workflow is a vehicle for operationalizing process integration and application interconnectivity.

Workflow demands performance from data administration and suggests a new paradigm as contained in Figure 10. This paradigm accommodates a broader view of business rules than previously taken by most data administrators.

Just as data administrators have shepherded data for the past two decades, workflow process administrators will begin to shepherd business processes. Process administrators have much to learn from the experience of their data administration colleagues, unless, of course data administrators want to take on the challenge of process administration.

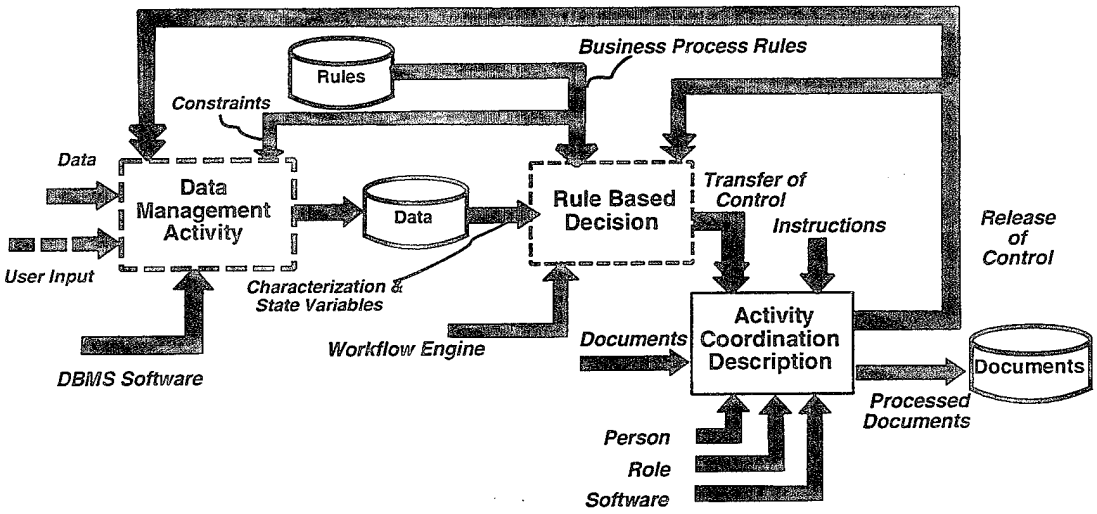


Figure 10 - An Enterprise Perspective

DONALD D. MARKS, Ph.D.
Richard S. Carson & Associates, Inc., 1 Skyline Tower, Ste. 2602, 5107 Leesburg Pike, Falls Church, VA 22041 703-379-5700 Fax:703-379-5707 Email: marks@carsoninc.com

Dr. Marks currently directs Business Process Reengineering, Workflow, and IRM service capability development for Carson Associates'. He is responsible for development of Model Solutions, Carson's methodology, techniques and tool set for analyzing enterprise operations and building solutions for the future. In previous roles at Carson Associates he facilitated the DoD Procurement CIM Activity Model and supported the extension of the DoD's Enterprise model.

Dr. Marks' previous experience includes BPR and Information Architecture consulting for the US Postal Service and Deutsche Bundespost. He also directed the development of NUS Corporation's BPR business approach and methodology. He has been an Assistant Division Manager for Stone & Webster Engineering where he directed an application development team which built information systems for electric utility clients; and lead a product team which built a document imaging and management capability. He has also managed the Information Services department for Niagara Mohawk's Nine Mile 2 Nuclear Power Station.

JAMES K. WASS, C.C.P.
Richard S. Carson & Associates, Inc., 1 Skyline Tower, Ste. 2602, 5107 Leesburg Pike, Falls Church, VA 22041 703-379-5700 Fax:703-379-5707 Email: wass@carsoninc.com

Mr. Wass is a Senior Analyst with Carson Associates where he develops and applies techniques for Business Process Reengineering, Workflow implementation and Electronic Meeting support. He has done Business Process Modeling and Business Process Reengineering under the Department of Defense CIM Initiative. He received an M.S. in Computer Systems Management from the University of Maryland.

INDEX OF AUTHORS

	PAGE
Anninos, CPT Dionysis, USA, U.S. Army Artificial Intelligent Center	173
Appleby, Ms. Betsy, DISA	53
Armacost, Maureen, Richard S. Carson & Associates, Inc.	*
Beersdorf, Mr. Jerry, Delfin Systems	141
Bosch, Mr. Christopher, The MITRE Corporation	241
Boyer, Lt. Col. K.D., USAF, Joint Command and Control Warfare Center	*
Bradley, Mr. James, Coleman Research Corporation	617
Burke, LTC John, DISC4 HQDA	37
Campbell, Mr. Ian, Wellic Limited	*
Carbone, Ms. Patricia, The MITRE Corporation	129, 155
Carlile, Mr. Brad, Cray Research, Inc.	603
Ceruti, Dr. Marion, NCCOSC RDT&E Div 4221	79
Coleman, Mr. James, Georgia Institute of Technology	505
Connolly, Mr. David, The MITRE Corporation	613
Cyanka, Mr. Philip, DISA-Center for Software	3
Dasher, Ms. Linda Rae, Richard S.Carson & Associations	*
Dutton, Ms. Barbara, James Martin Government Intelligence, Inc.	491
Elliott, Ronald, HQUSMC	*
Emrich, Dr. Marco, Cincom Systems, Inc.	167
Esch, Jan-Marie, County Sanitation Districts of Orange County	197
Ficklin, Ms. Susan, The MITRE Corporation	613
Fisher, Ms. Donna, NCCOSC	185
Fleming, Ms. Jeanine, Richard S. Carson and Associates, Inc.	623
Fortier, Dr. Paul, University of Massachusetts - Dartmouth	185
Frieder, Dr. Ophir, George Mason University	117
Glass, Mr. Carter, Rapid Systems Solutions, Inc.	341
Gleicher, Ms. Andrea, USAF Phillips Laboratory	311
Gressang, Mr. Randall, SWL Division, GRC International, Inc.	99
Haigh, Mr. Tom, Secure Computing Corporation	443
Harris, E., SWL Division, GRC International, Inc.	99
Harris, Mr. Ron, SRA Technical Services Corporation	319
Henderson, Ms. Lynn, DISA	359
Hermansen, Dr. John, Language Analysis Systems, Inc.	593
Hicks, Ms. Lila, The Aerospace Corporation	311
Hildenberger, Ms. Ruth, The MITRE Corporation	155
Hillman, Mr. Tom, Delfin Systems	141
Hopkins, Mr. Michael, HQ United States Central Command	*
Hufford, Mr. Duane, American Management Systems	219
Hughes, Mr. David, Dbx, inc.	185
Huo, Dr. Chien, Defense Modeling and Simulation Office	15
Jones, Ms. Tanya, Decision Systems Technologies, Inc.	585
Kamel, Dr. Magdi, Naval Postgraduate School	79
Kameny, Iris, The RAND Corporation	15

Kepler, Ms. Feliza, Data Networks Corporation 359
Kerchner, Dr. Marcia, The MITRE Corporation 155
King, Ms. Amy, Decision Systems Technologies, Inc. 585
Klauder, Mr. L. Tobias, Vector Research, Inc. 71
Koltz, Mr. Mark, Decision Systems Technologies, Inc. 585
Lee, Mr. Richard, Sterling Software, ITD 593
Lefler, Mr. Mike, PRC, Inc. 409, 617
Levene, Mr. Neal, Vector Research, Inc. 71
Little, Ms. Jennifer, Amerind, Inc. 205
Lu, J., SWL Division, GRC International, Inc. 99
Magee, Mr. Peter, NCI Information Systems, Inc. 527
Marks, Dr. Donald, Richard S. Carson & Associates, Inc. 631
Martin, Dr. Michael, DISA/JIEO/Center for Software 417
Mathwick, J., SWL Division, GRC International, Inc. 99
Mattox, Dr. David, The MITRE Corporation 155
McHenry, Ms. Bonnie, NCI Information Systems, Inc. 527
McNeilly, Mr. Chris, Sterling Software, ITD 593
Michaels, G., SWL Division, GRC International, Inc. 99
Morse, Dr. H. Stephen, MRJ, Inc. *
O'Brien, Mr. Dick, Secure Computing Corporation 443
Osterholtz, Ms. Louise, Sterling Software, ITD 593
Otano, Mr. Hernan, Richard S. Carson and Associates, Inc. 465
Paolicelli, Mr. Dave, Vector Research, Inc. 71
Paul, Ms. Alta, DISA-Center for Software *
Piper, Ms. Pamela, DISA-Center for Software 351
Quigley, Ms. Gail, AT&T Global Information Solutions 297
Reilly, Ms. Diane, Richard S. Carson and Associates, Inc. 623
Renner, Dr. Scott, The MITRE Corporation 107
Roark, Mr. Mayford, Martin Marietta 185
Rosenthal, Mr. Arnon, The MITRE Corporation 107
Ruocco, Major Anthony, USA 117
Safran, Mr. Larry, Delfin Systems 141
Satterwaite, Mr. Darin, The MITRE Corporation 613
Scarano, Mr. James, The MITRE Corporation 107
Schuringa, Mr. Tjakko, Covia Technologies Europe *
Sills, Ms. Lisa, Georgia Institute of Technology 505
Smith, Dr. Kenneth, The MITRE Corporation 129
Smith, Ms. M. Cassandra, The MITRE Corporation 613
Stanford, Ms. Karen, Decision Systems Technologies, Inc. 585
Stickel, Mr. Dan, Delfin Systems 141
Stokrp, Mr. Mark, PRC, Inc. 409
Tharpe, MAJ Leonard, USA, U.S. Army Intelligent Center 173
Thompson, Mr. David, Acucobol, Inc. 539
Thomsen, Mr. Dan, Secure Computing Corporation 443
Thuraisingham, Dr. Bhavani, The MITRE Corporation 1, 79
Timblin, Ms. Barbara, Symnatec Corporation 515
Turner, Mr. Michael, The MITRE Corporation 129

Uhrig, Mr. Rick, Sybase, Inc.	457
Valentine, Mr. Peter, U.S. Army Electronic Proving Ground	*
Vandivier, Mr. Stephen, Avanco International, Inc.	331
Verga, Mr. Andrew, Wizdom Systems, Inc.	53
Waggoner, Mr. Duane, Commander Naval Security Group	27
Waldron, 1Lt Lee, USAF, Automate Communications Systems	251
Wass, Mr. J.K., Richard S. Carson and Associates, Inc.	631
Wineland, Ms. Joyce, Office of Naval Intelligence	*
Woody, Ms. Ann, DISA-Center for Software	71
Worthington, Mr. Jess, Informix Federal	367

*Paper not available at time of printing; however, we do expect it to be presented at the Colloquium.