STRATEGY RESEARCH PROJECT

DATA INTEROPERABILITY FOR SYSTEMS OF SYSTEMS: OUR ACQUISITION PARADIGM MUST CHANGE TO ACHIEVE IT

BY

LIEUTENANT COLONEL WILLIAM T. LASHER
United States Army

USAWC CLASS OF 1997

U.S. ARMY WAR COLLEGE, CARLISLE BARRACKS, PA 17013-5050

USAWC STRATEGY RESEARCH PROJECT

DATA INTEROPERABILITY FOR SYSTEMS OF SYSTEMS: OUR ACQUISITION PARADIGM MUST CHANGE TO ACHIEVE IT

by

LTC William T. Lasher

COL Donald R. Yates
Project Advisor

U.S. Army War College
Carlisle Barracks, Pennsylvania 17013

ABSTRACT

AUTHOR: William T. Lasher (LTC), USA

TITLE: DATA INTEROPERABILITY FOR SYSTEMS OF SYSTEMS: OUR ACQUISITION PARADIGM MUST CHANGE TO ACHIEVE IT

FORMAT: Strategy Research Project

DATE: 16 April 1997 PAGES: 38 CLASSIFICATION: Unclassified

To achieve the information superiority called for in Joint Vision 2010, we must start building information systems which can freely exchange data and be networked into systems of systems. Unfortunately, DOD's current system acquisition paradigm, which stresses independent development of systems, makes this goal difficult or impossible to achieve. To realistically achieve a shared data environment in which systems of systems can be built, we must change the acquisition paradigm. DOD needs to undertake a major development program to build and maintain a quality enterprise data architecture. This architecture would provide a foundation upon which future information systems would be built. It would ensure data interoperability and could ultimately provide the basis for a set of integrated corporate databases across DOD.

TABLE OF CONTENTS

TABLE OF FIGURES

Introduction

Success of the military after next will clearly be heavily dependent on our ability to adeptly manage information. Pick up any of our recently published vision documents and this fact virtually jumps out. Army Vision 2010 calls for us to "Gain Information Dominance… to create a disparity between what we know about our battlespace… and what the enemy knows about his."[1]

Joint Vision 2010 foresees "…increased access to information and improvements in the speed and accuracy of prioritizing and transferring data brought about by advances in technology.… We must have information superiority: the capability to collect, process, and disseminate an uninterrupted flow of information while exploiting or denying our adversary's ability to do the same."[2] It calls for us to develop "…a new conceptual framework for operations. The basis for this framework is found in the improved command, control, and intelligence which can be assured by information superiority."[3]

To attain this information superiority, we will have to do much more than buy new hardware and develop advanced software. We will need to build new systems which can be networked together so they can freely interoperate. In essence, we will need to build systems of systems. DOD's current acquisition paradigm, however, does not enforce or support development of interoperable systems. We must change our method of acquiring information systems to achieve the interoperability necessary to develop systems of systems.

In this paper I will discuss the types of interoperability necessary to create a system of systems. I will show why the current acquisition system severely inhibits achieving data interoperability necessary for the realization of this goal. Finally, I will discuss alternatives to the current acquisition strategy that could provide the type of interoperability which facilitates development of joint systems of systems.

The Need for Systems Of Systems

In his visionary article, "The Emerging System of Systems" Admiral William Owens describes a future battle environment where "systems of systems" will synergistically improve the strategic leader's abilities to command and control joint forces. They promise to keep commanders at all levels fully informed, assist them in better and timelier decision making, and, in some cases, automatically detect and respond to events; a feat largely beyond our grasp today.

So, what is a system of systems? In essence, it is an executive level automated system which pulls data from functional level information systems (IS). (The classic Army functional systems support the battlefield functional areas and include systems such as AFATADS, CSSC2, ASAS, etc.) As shown in Figure 1, the executive information system could poll subordinate information systems for either raw (base level) data, or some form of aggregate or abstract data derived from the subordinate system's base level data. Subordinate information systems could also be programmed to pass critical data up to the executive system periodically or based on key events. The executive level system could then present this information to senior decision makers in some form to assist him/her in making decisions.

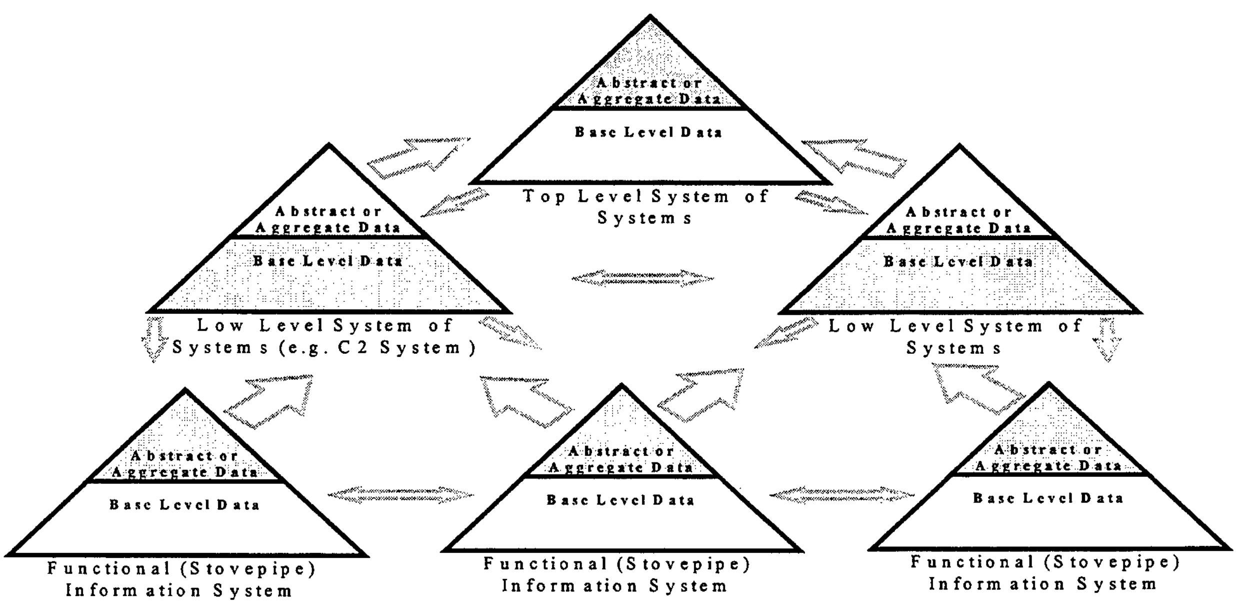


Figure 1, Illustration of a System of Systems

In some predetermined cases, executive level systems could even instruct subordinate information systems to take action, based on an automated analysis of the information it has received; for example, detection of a missile launch.

Actually military systems of systems have existed for centuries. A standard command and staff structure is essentially a system of systems. Subordinate commanders and staffs freely communicate laterally. They provide information and recommendations to a senior commander, and, based on his interpretation of the information, the commander provides guidance back. In many cases today, while we have automated functional information systems which assist staff officers and commanders, the interface between these systems is still a human.

In a true system of systems, as Admiral Owens envisions it, data would be freely passed between functional and executive level information systems without requiring human interpretation or intervention. It is this total interoperability between systems which will ultimately allow us to drastically improve battlefield awareness and shorten our decision cycles.

Considerations in Building Interoperable Systems

Three primary challenges must be overcome to allow any two systems to "talk" to each other directly (see Figure 2). First, the systems must be technically compatible; that is, system A must have a communications interface electronically compatible with that of system B. Second, a communications link must be established between the systems. Third, system A must correctly interpret the information it gets from system B.

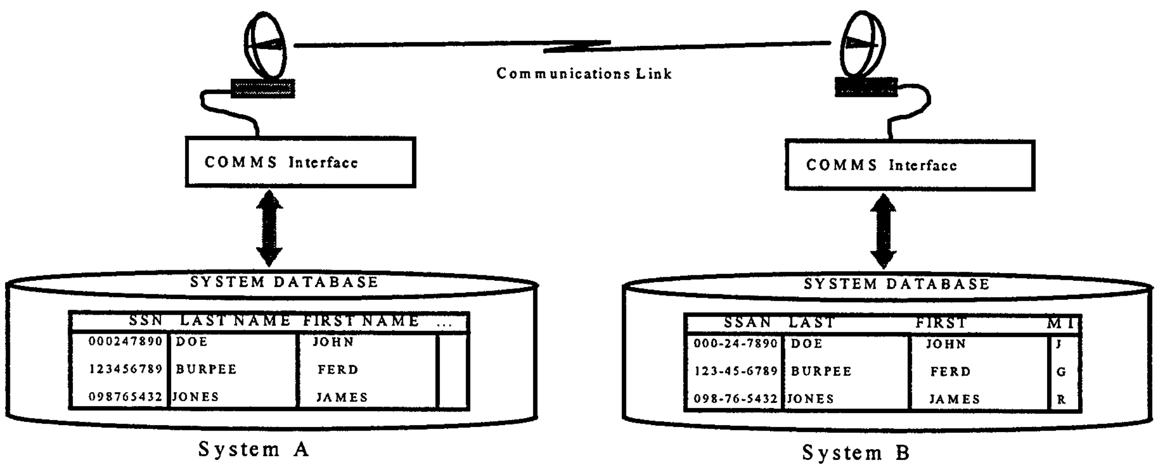


Figure 2, Communicating Between Systems

The Technical Challenges

The first two "technical" challenges can be solved, given the right hardware, software, and technical expertise (i.e. given enough money). Modern network technology and maturing industry standards (such as those used for the internet), are making the technical problems far less formidable than they once were.

The Data Challenge

The third, and most difficult, challenge in allowing systems to talk to each other is getting them to exchange data. This is actually a design problem. It can be difficult, or impossible, to properly exchange data between systems which have different data designs.

A Short Primer on Data

To properly understand the third challenge, we must define some terms and constructs commonly used in the information management community.

Data is defined as a "representation of facts, concepts, or instructions in a formalized manner suitable for communication..."[4] The character string *071241*, for example, could represent some random piece of data. Note that in and of itself, the data has no particular meaning. Only when we associate the data with a *context* or *meaning* does it become valuable. If we associate context with this character string, such as: *DATE(ddmmyy)=071241*, the data starts to take on meaning. (In this case we know the character string represents December 7, 1941).

A ***data element*** is "a basic unit of information having a meaning and subcategories of distinct units and values."[5] The example used above; *DATE(ddmmyy)=071241* is a simple form of a data element.

A ***data model*** (also called an *information model*) is a model which graphically shows the "things" (called *entities* in infospeak) an organization manages and how they relate to each other. Each *entity* (thing) has characteristics (called *attributes* in infospeak) that give it its identity. These *attributes* (characteristics) form the basis of data elements. The data model provides additional context by tying together those attributes which relate to a particular entity. This grouping of attributes forms the eventual basis for *records* (or *tuples* in infospeak) in a database. Figure 3 illustrates these constructs.

A data model can be an exceptionally powerful tool. In essence, it provides a high level design specification for a database. In fact, there are computer aided software engineering (CASE) tools available today which will create database designs directly from data models.

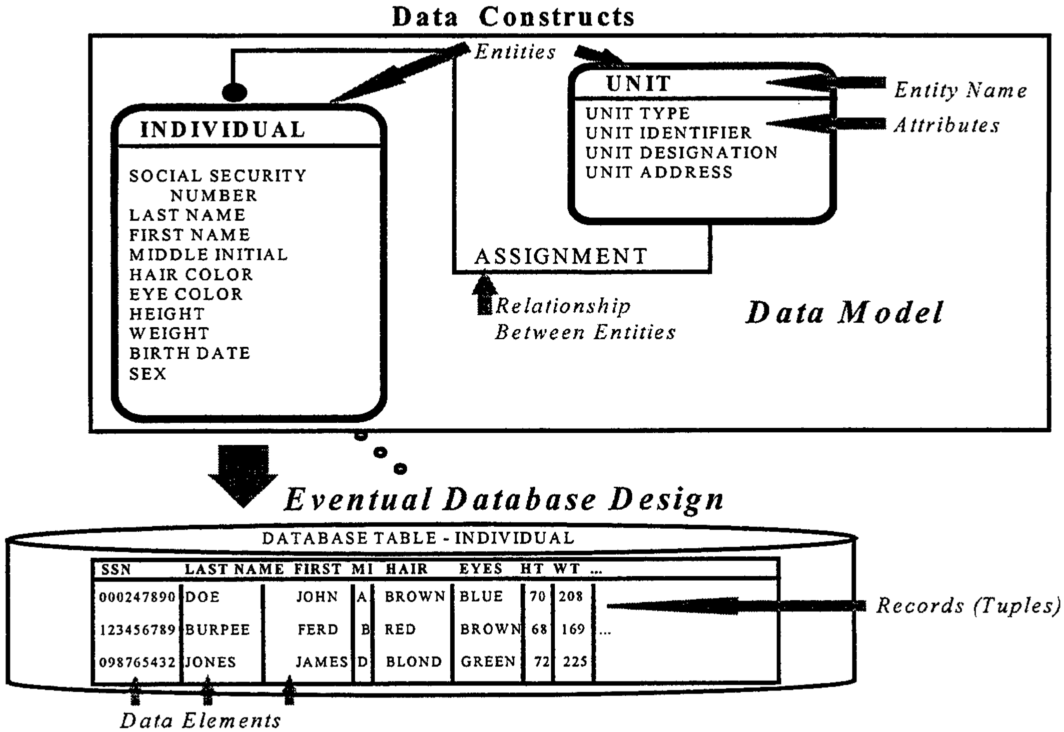


Figure 3, Illustration of Common Data Constructs

A ***data architecture*** is "the framework for organizing and defining the interrelationships of data in support of an organization's missions, functions, goals, objectives and strategies."[6] Basically, a data architecture is a high level data model of the entire enterprise.

The term ***data infrastructure***, as used below, is the actual implementation of an enterprise's data architecture. The Assistant Secretary of Defense for C3I refers to the need to "work towards a corporate integrated data base that enables true sharing of information and data."[7] This corporate integrated database would, essentially, be a data infrastructure for DOD.

Data Interoperability

Now that we are familiar with the basic concepts behind data, we are equipped to discuss the data interoperability problem, or the challenge in getting systems to exchange data.

Automated information systems universally store information in large databases made up of individual data elements. The format and meaning of each data element is rigorously defined during information system development and is fundamental to the system design. Development of an information system's database design is a large project which typically constitutes a major portion of the overall system development effort.

To accurately transfer an element of data from system A to system B, two requirements must be met: First, the data must be presented in the form that system B expects; second, the context of the data must be common between both systems. It is generally feasible to translate data from one format to another. (Although this requires development of new, and often expensive, software to do the translation.)

It can be difficult, or even impossible, however, to accurately translate data defined in one context to that defined in another. (The apples to oranges analogy applies here). Hence, two independently built information systems may not be able to share data, if they fundamentally differ in the way their data elements were defined during systems design.

The Growing Need for "Sharable" Data

DOD has long recognized the need for building systems with sharable data. In the stovepipe era, however, information systems were built primarily to perform one and only one function (the Joint Uniformed Military Pay System - JUMPS, for example). These were large, self-contained systems with massive databases run on mainframe computers from a central location. Development of each legacy system was a major effort and was tightly controlled. (The "waterfall" development model, discussed below, was used to build systems of this era.)

This lockstep method of system development tended to ensure that data design was consistent (thus data was sharable) within that large system. The need for sharing data across these large "legacy" systems, while important, was not critical since each generally performed a different and completely independent function.

As computer technology has advanced, computers have become increasingly smaller and more powerful. We are moving away from the centralized mainframe environment to one which is highly distributed. Advancing technology is effectively removing a discipline previously imposed by the size and expense of mainframe computers. Development efforts are now smaller and much less tightly controlled.

We are seeing the appearance of multiple systems built at different echelons which perform similar functions and track similar (sometimes even the same) data. The focus of each is usually exclusive to the one function it is to perform at the expense of interoperability. Furthermore, these systems are often built by contractors who have virtually no interest in making them interoperable with other systems.

In this new distributed environment it is becoming absolutely critical we design information systems such that they can share data. As we saw above, our senior leadership is starting to recognize the extraordinary potential we could realize if systems could be networked together and freely exchange information.

Leaders of our technical organizations have established development of a shared data environment as one of their principal goals. The 1993 Army Enterprise Strategy specifically mandates that "All information systems will use Army standard data elements. This will increase the accuracy and timelines of the data, increasing interoperability during all operations."[8]

LTG Edmunds, Director of the Defense Information Systems Agency states, "There is no greater imperative than to deliver to Warfighters fully integrated systems that provide [a] fused, real-time, ground truth picture of the battlespace."[9] The goal is clear and relatively simple. Developing a method to achieve it is another matter.

The Flaw in Our Current Acquisition Paradigm

Why is building information systems that share data so hard? A great deal of the reason has to do with the way we acquire them. DOD and service information systems are built using the standard DOD acquisition model. Each major system is, for the most part, developed independently by a program manager who is provided reasonable autonomy and held responsible for progress in system development and fielding. The program manager's primary motivation is delivery of a system on time and within budget. While (s)he undoubtedly desires to achieve interoperability with other systems, there is little hope (s)he can coordinate the system design with every other system (existing or under development) that may someday interface with hers/his.

DOD Funding mechanisms also focus narrowly on independent systems. As Admiral Owens points out, "We have cultivated a planning, programming, and budgeting system that tends to handle programs as discrete entities. The PPBS cycle forces us into a compartmentalized perspective."[10]

Thus, DOD's acquisition system is really designed to optimize the efficiency and effectiveness of individual systems. Unfortunately, it does so at the expense of developing (or even allowing the development of) systems of systems with their promised synergistic performance.

The Method : How we Build "Watches" Now

To illustrate why we are where we are (having spent billions of dollars constructing sophisticated information systems which, for the most part, do not interoperate), we need to discuss how we design and build information systems.

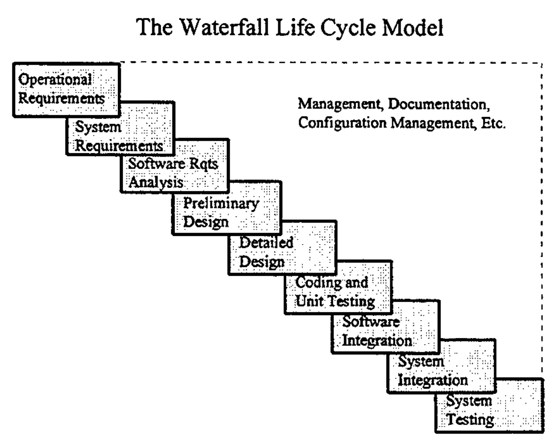


Figure 4 The Waterfall Model

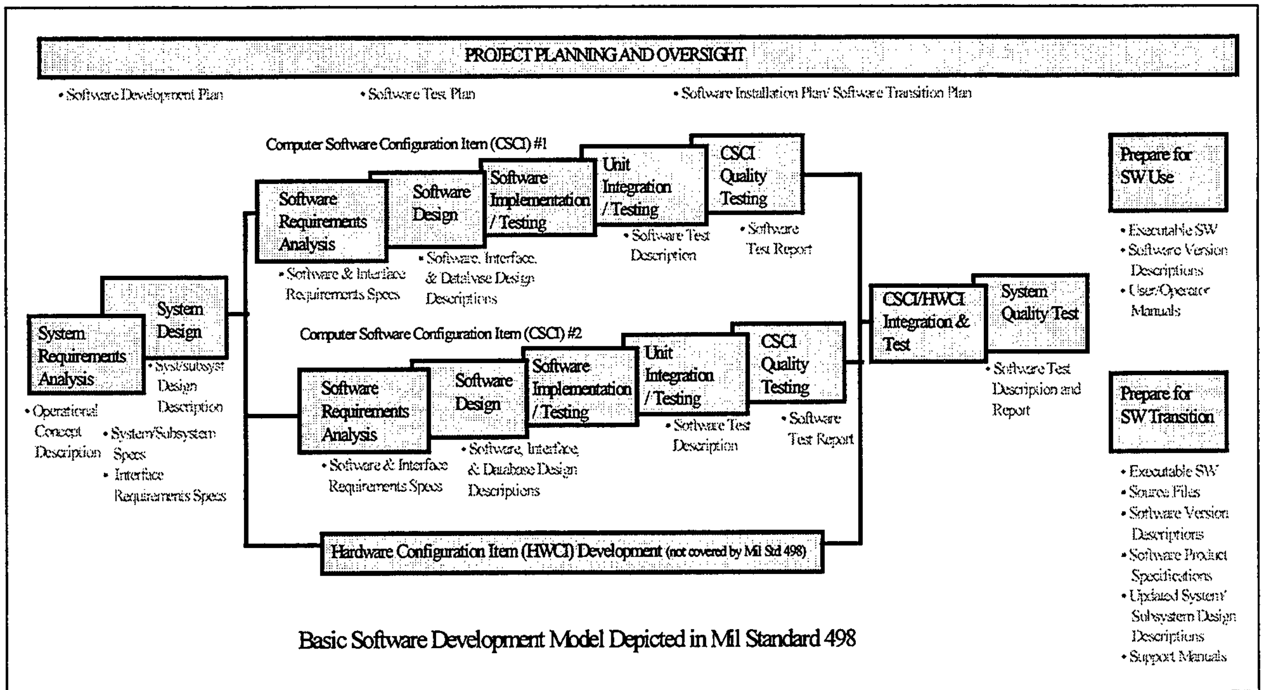


Figure 5 Mil Standard 498 Software Development Cycle

Figure 4 shows the typical "waterfall" development model used to build software during the mainframe era.[11] Figure 5 shows the newest software development lifecycle model approved as part of DOD's Mil Standard 498. Both are process, or function, centered models. If one envisions a pie representing all functions performed across the services, these models take a slice of that pie and automate the functions within (possibly a very small part of) that slice (see Figure 6). A portion of the development effort will involve designing the system database -- or building a system data architecture.

Take the Standard Installation/Division Personnel System (SIDPERS), for example (or substitute your favorite system here). Under these models we would conduct a thorough analysis of all the processes involved in managing personnel at divisions and installations. Based on that analysis, we would develop a set of requirements to automate these processes, a data architecture for the system, and ultimately a detailed design to automate these processes.

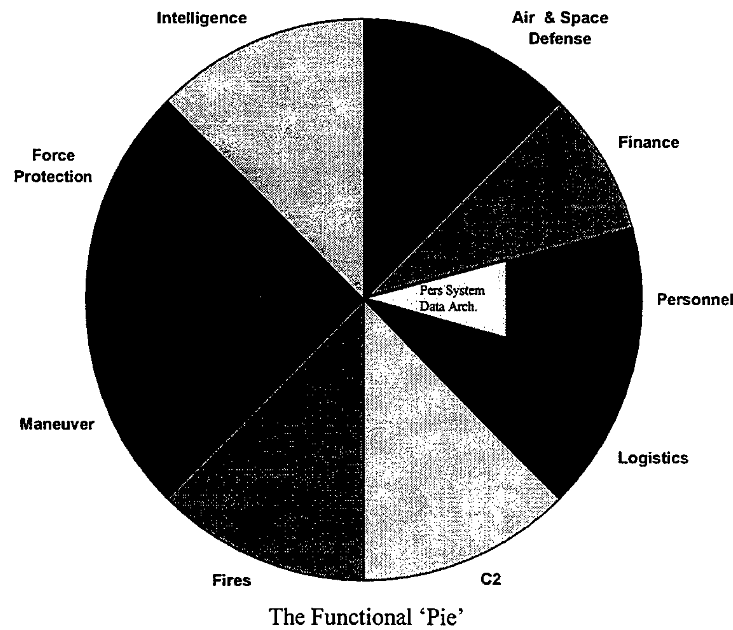


Figure 6, Example Functional Partitions

The Problem With the Single System Focus

The entire system from its genesis is narrowly oriented around the slice of pie involving one particular function(in this case personnel management) and inherently drives the system developer into a stovepipe. Effectively, once the system developer is given his charter ("Go forth and automate function xyz."), our system development models call for him to concentrate within that narrow lane. There is no construct in the formal models that causes the PM to look outside his lane and integrate his system with others. There are no provisions in these models which compel a system developer to design interoperability into the system. In fact Mil Standard 498, just over two years old, does not even mention interoperability of data. Thus interoperability, including the ability to share data with other systems, is typically handled as an adjunct to building the basic system.

If the PM strictly followed the formal system development models, (s)he might well have fully developed the system's data architecture before even considering interoperability. As we saw above, however, systems designed and built independently can have substantially different data designs, and thus, may not be capable of sharing data.

Options for Achieving an Interoperable Data Environment

There are at least three general courses of action DOD could pursue in developing systems which could freely exchange data. It could centralize all systems development efforts under one organization within DOD. It could continue to allow decentralized development while insisting developers adhere to strict interoperability standards. Or, it could change the acquisition method by making system development a joint effort between the system developer and an organization responsible for development of an enterprise data architecture. Each of these options is discussed in more detail below.

Centralize: Develop Systems only at the DOD/Joint Level

This method follows the "massive centralization" train of thought. Under this course of action, we would pull all system development effort and expertise to a central department under DOD or CJCS. This agency would be responsible for development of all new information systems within DOD. It would implement rules and procedures to ensure systems were developed such that they maintained interoperability.

The obvious trouble with this approach is it becomes totally unresponsive to the needs of the field. It also tends to promote the development of massively large "do everything for everyone" systems which are exceptionally complex and difficult to build. Furthermore, if we continued using the existing acquisition paradigm (independently developed systems), all we would have done is move the problem to a higher level. There is no guarantee systems developed by a large centralized agency will be more interoperable than any others.

Standardize: Remain Decentralized but Build and Enforce Standards

DOD has commonly called this the "data standardization program." It is the course of action which both DOD and the Army have been pursuing in some form for at least the last thirty years[12]. The persistent and widespread lack of interoperability within DOD systems[13] [14] would seem to indicate something about either this course of action, or the way we are pursuing it, is just not working.

The Concept

Data Standardization calls for development and implementation of technical and data interoperability standards to which system development efforts would be held. Data standards are centered around an enterprise data architecture (data model) and standardized definitions of data elements called "standard data elements." These are kept in a repository, or dictionary, which would be universally available to system developers.

In theory, system designers could go look in the dictionary and pull out the "standard" definition for any DOD data and use that in their design. Under the current guidance,[15] if the developer does not find a suitable standard to use, (s)he is then responsible for developing a *proposed* standard and submitting it to the DOD Data Administrator for approval. In this manner, the DOD enterprise data architecture is supposed to be developed over time as new systems are built.

The Fallacy of Standard Data

The word "standard" evokes an image of a set of rules, protocols, or specifications which rarely change over time and need little periodic maintenance. Unfortunately, construction of an enterprise data architecture is a massive project requiring significant development effort and considerable upkeep.

We stated above that database design is a major portion of the development effort in building any given information system. Developing an enterprise data architecture is, in essence, the construction of a high level data design for every functional area in the enterprise. It is more an engineering effort than one of developing a standard. And while a system's data architecture is relatively fixed compared to other system components, it can change over time. Thus calling it a standard can be deceiving.

The implication behind the data standardization program is that at some point the data architecture will be "finished," and can be placed in caretaker status. However, experience during the Army's data modeling efforts in the early 1990s showed that as new functional areas were modeled, we discovered inconsistencies, oversights, and errors in the existing data architecture. It is reasonable to assume that as long as new design work is ongoing (certainly for the foreseeable future), this will continue to occur. To ensure the enterprise data architecture is correctly maintained will require continual refinement and some sort of quality control over the process.

Cost , Complexity and Quality

The existing Army and DOD data standardization programs are really attempts to develop an enterprise data architecture with minimal investment, and have seriously underestimated the effort required. These programs have consistently been woefully under-funded and understaffed, and promise little hope of developing, or maintaining, a quality enterprise data architecture.

These programs have also vastly underestimated the complexity involved in building and maintaining an enterprise data architecture, especially for an organization as large and diverse as DOD. The current DOD Data Model, which is relatively young, has 3453 entities with another 5000 under development. The DOD Data Dictionary System, which is used to store DOD standard data elements, has 23,658 elements approved, proposed, or under development to date.[16] Obviously, as future information systems are developed, the data model and the number of standard data elements will grow.

Under the data standardization method, the system designer (usually a contractor) must become familiar enough with the DOD data architecture and DOD standard data to develop a database design incorporating these standards. At the very least, this requires a significant effort to learn the intricacies of these complex standards. At worst, (while standards are still not fully developed) the system developer will find few or no applicable standards and will be required to develop them. Either way, the database design team would spend an inordinate amount of time "spinning up" on the DOD standards. This one step would unquestionably add considerable cost to any given system development effort.

As with most engineering products, the utility of any data architecture is highly dependent on its quality. If it fails to accurately represent the entities and business practices of an enterprise, it will not support construction of useful information systems. Unfortunately, once a data architecture is defined and systems are built to its specification, it becomes an expensive proposition to change the architecture upon discovery of an error. Thus, development of a quality and accurate data architecture from the start is crucial.

Determining the correct entities, relationships, and business rules for a large data architecture is an exceptionally difficult mental drill. Managers who participate in data modeling sessions often find themselves rigorously defining their business practices and realizing they have never really done so before.

A program manager, whose primary motivation is delivery of a system, is unlikely to take the care desired in developing his/her portion of the enterprise data architecture. The potential with this method of developing an enterprise architecture, is that we will evolve a product whose quality is very suspect.

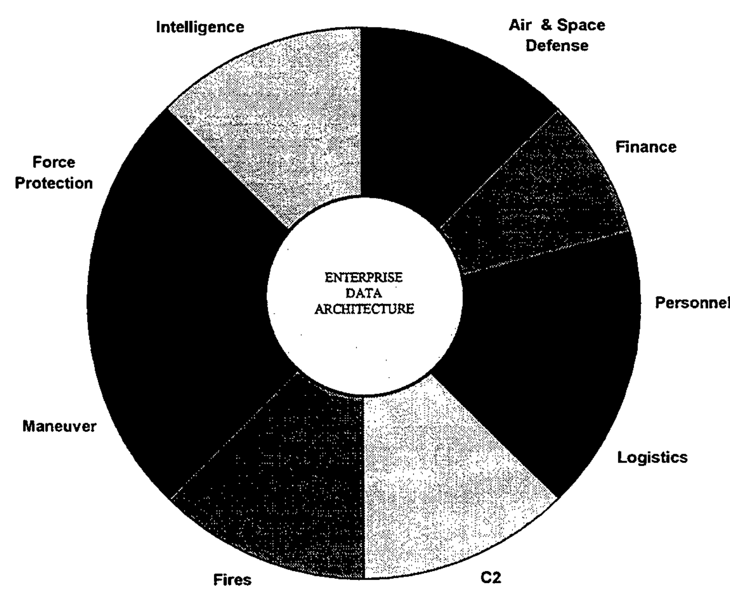
Change the Acquisition Process

Another possible course of action is to modify the DOD Acquisition paradigm and structure. This course of action splits development strategy for information systems into two parts.

Development of the data architecture would be done centrally, while allowing the remainder of system development to stay decentralized.

The Concept

Under this approach, the enterprise data architecture would be viewed as a major engineering project, not a set of standards. We would fund, build, and maintain a DOD data architecture as a major system development effort. Unlike most development projects, however, the product would not be a system designed for end users. It would, instead, be a system built exclusively to support other information system development efforts. In essence, this approach advocates construction of DOD's enterprise data architecture as a large infrastructure project which provides a foundation upon which other (end user)



Data Architecture at the Core of Systems Development

Figure 7, Enterprise Data Architecture

information systems are built (see Figure 7)

A necessary step in this process would be development of a comprehensive information system designed to support construction of the data architecture. This system would be a Computer-Aided Software Engineering (CASE)-type tool designed to assist users in navigating and modifying the data architecture. It would also assist system developers in incorporating the architecture into new information system design.

This approach would also recognize the inevitable need to maintain the data architecture over the long term. An organization's data needs and business practices will change (usually slowly) over time. If the data architecture doesn't change with the organization, it becomes obsolete and ultimately useless.

To retain its utility, the architecture would have to be modified periodically. This modification must be closely controlled to ensure components of the architecture (models, data elements, etc.) remain consistent. Mechanisms must also be built which eventually cascade changes in the enterprise data architecture down to existing information systems.

The Organization

Under this approach, we would charter a high level organization in OSD or JCS to centrally develop and maintain DOD's data architecture. This organization would also be charged with assisting information system developers in using the enterprise data architecture to design and build new systems. A proposed organization appears at Figure 8.

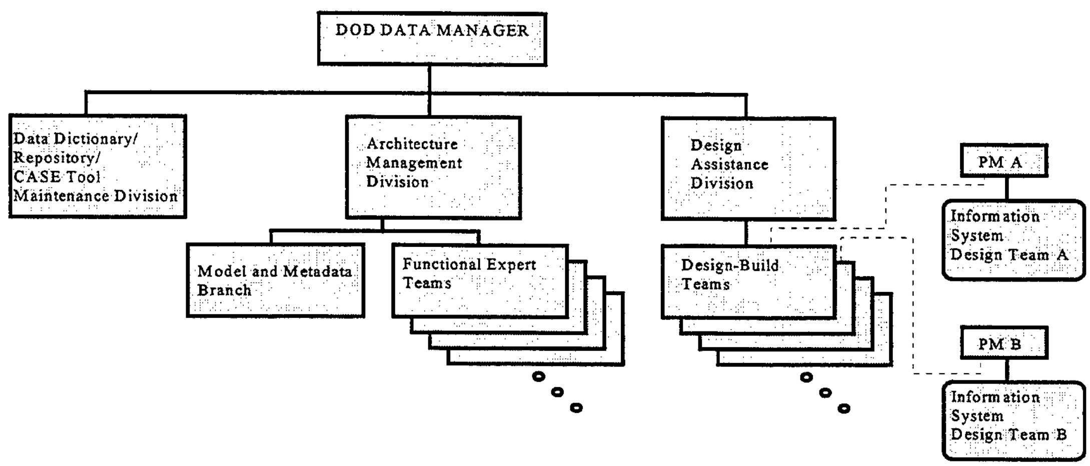


Figure 8, Possible Layout for DOD Data Management Organization

The Dictionary/Repository division would be responsible for maintaining the information system (CASE tool) in which the architecture is kept. The Architecture Management Division would continually update and maintain the architecture to ensure its currency, quality, and consistency. Teams of functional experts would be responsible for portions of the architecture that fall into their particular functional area.

The Design Assistance Division would comprise several design-build teams. Each design-build team would work with an individual IS developer during system development on a dedicated basis. The team would assist the IS design team in database specification, design, and development. That same team would also monitor and assist the PM as needed on all database redesign issues through the entire lifecycle of the program. Design-build teams could call on other parts of the organization, such as functional expert teams, to assist with development efforts.

Building the Architecture

Construction of the architecture would clearly be a massive job in itself. It could, however, be done incrementally given the right organization and a consistent funding stream.

The design-build teams would actually perform two functions. First, as stated above, they would assist the IS developer in his database design and system development. Second, they would incrementally build parts of the DOD enterprise data architecture.

As past experience indicates, in the course of developing the data design and specifications for its assigned information system, a team would likely discover omissions, errors, or inconsistencies in the existing enterprise data architecture. The team would then be responsible for engineering a new portion of the enterprise architecture required to support its particular development effort.

As design teams developed additions or corrections to the architecture, the Architecture Management Division could integrate them into the enterprise architecture, ensuring they remained consistent with existing portions. As the enterprise architecture matured over time, design-build teams should find fewer omissions and errors; thus development would take less time and effort.

While this approach to building an enterprise architecture is similar to the approach we are currently pursuing under the data standardization program, it differs in that only the DOD Data Manager is responsible for the architecture. The DOD Data Manager's focus is primarily on development of a quality and consistent enterprise architecture. The PM, on the other hand, can focus on building a system without having to devote his/her resources toward building the enterprise architecture (a job which (s)he has little motivation or expertise to do).

Advantages and Disadvantages

This approach could have several advantages over our current standardization approach. As stated above, it removes the burden of developing an architecture from the PM and places it on an organization designed and staffed to do that job. Design-build teams would be fully familiar with the enterprise architecture, and thus would require little "spin-up" before they start working. Their familiarity with the enterprise architecture means they could immediately take advantage of existing designs in the architecture and apply them to the development effort. It also means they could quickly identify omissions, inconsistencies, or errors in the enterprise architecture and work to get them corrected. Finally, given that one organization would be responsible for the data architecture, its quality, consistency, and integrity should be considerably better than one developed by multiple organizations.

There are clearly some tough issues that must be addressed with this "team" approach to information systems development, however. The fundamental change from the Program Manager's point of view is that (s)he would no longer have exclusive control over the database design team. Database design would, instead, be a joint effort between the program manager's office and a DOD design-build team.

Before the enterprise architecture matures, the design-build team's activities could slow development efforts as the team integrates newly designed data models back into the enterprise architecture. (Comparatively, however, this would not slow down a project nearly as much as our current system which calls for the PMO to do much of this same work.)

Design-build teams would initially require time to become familiar with the specific project. There is no reason to believe that they would require significantly more time than any normal development team starting a project to do this, however.

The most difficult issues would likely be in the contracting arena. Effectively, we would be requiring a contractor to work with a government provided team for a portion of the development effort. Since the government team's activities could either slow down or speed up (depending on the maturity of the enterprise architecture) development efforts, writing a contract fair to all parties would be a challenge. This would be even more difficult if design-build teams were, themselves, partially or wholly staffed with contractors.

Despite these challenges, this approach offers considerable promise. It explicitly recognizes the need to undertake a major infrastructure type project to build and maintain a quality enterprise data architecture. It provides for an organization to do so. It provides tools and personnel to assist the system developer in building new information systems. And it promises true DOD-wide data interoperability and potential long term cost savings.

Potential Benefits of an Interoperable Data Environment

DOD-wide data interoperability, in turn, would provide a common shared data environment across DOD. The potential benefits of such a common data environment are extraordinary.

Systems compatible with the DOD enterprise data architecture could, in theory, freely pass data between themselves without translation and with assurance that definitions behind the data are common. This "complete interoperability" would make it possible to build systems of systems without having to modify the underlying functional information systems and without having to build translators.

A fully developed DOD data architecture also promises to eliminate significant portions of individual system development efforts since much of the database definition within any functional area would already exist. In fact, given new Computer Aided Software Engineering tools, one could envision database design being done by merely selecting the desired entities, relationships, and attributes from an already-constructed DOD data model.

A shared data environment would help eliminate the growing proliferation of redundant information systems. Today, different services and organizations within them have developed different and incompatible systems which often functionally overlap to some extent. They maintain the same data, but in different form (How many times have you provided the same personal data to different agencies for their different databases?) Interoperable systems which require the same data could pull that data from other (parent) systems rather than requiring duplicate data entry. Better yet, common data definitions across all services would allow us to eliminate systems which duplicate the functions of others.

This shared data environment would also facilitate development of truly reusable software. Both the Army and DOD have long pursued a goal of establishing a repository to maintain reusable software modules. This goal has eluded them largely because software operates on data; and if two systems design their data definitions differently, they generally can't use the same software. Interoperable data promises to make reusable software a viable possibility.

The ultimate goal for an enterprise data architecture could be the development of an integrated system of functional on-line databases. Given the near universal accessibility that internet technology provides, developing an information system in the future could be no more complicated than forming a series of queries against these already existing databases.

Conclusion

The potential advantages that integrated systems of systems offer truly are synergistic. Unfortunately, our current acquisition model inhibits the development of systems which can freely share data and interoperate. If DOD is to develop interoperable systems, we should fund and undertake a major development effort to build an enterprise data architecture. We must staff an organization of experts responsible for the maintenance of this architecture. Further, we should alter our acquisition model such that database design and development occurs jointly between the program manager's office and the organization responsible for the DOD enterprise architecture

In the words of the Honorable Emmett Paige, ASD(C3I), "...information that is part of a shared integrated information database, accessible by a wide user base that can collaborate, has tremendous value. The rapid pace of technological advance, coupled with an unpredictable world situation demand that we pursue this goal with all deliberate speed."[17]

ENDNOTES

[1] Department of the Army, Army Vision 2010 (Washington, U.S. Department of the Army, 1996), 17.

[2] Department of Defense, Joint Vision 2010 (Washington, U.S. Department of Defense, Chairman of the Joint Chiefs of Staff, 1996), 16.

[3] Ibid., 19.

[4] Department of Defense, DOD Data Administration, DOD Directive 8320.1 (Washington: U.S. Department of Defense, 26 September 1991), Encl 2, para 2.

[5] Ibid., para 8.

[6] Department of Defense, Data Administration Procedures, DOD 8320.1-M (Washington: U.S. Department of Defense, 29 March 1994), ix.

[7] Emmett Paige Jr. (HON), "Achieving the Integrated Systems Concept," Keynote address at the Armed Forces Communications and Electronics Association International Technet '96 Convention, Washington, 4 June 1996, as published in Defense Issues 11 no. 51 (June 1996): 2.

[8] Department of the Army, "Army Enterprise Strategy, The Vision," (Washington: U.S. Department of the Army, Office of the Secretary of the Army, DISC4, 20 July 1993): 22.

[9] Defense Information Systems Agency Memorandum for Program managers and developers, "Implementing DOD Standard Data Elements", (April 1996): 1.

[10] William A. Owens, ADM, "Emerging Systems of Systems," U.S. Naval Institute Proceedings 121 no. 5(May 1995): 36.

[11] Paul D. Condit, MAJ, USAF, The model shown in Figure 4 is taken from the research report Principles of Information Resource Management A Foundation for the Future (Maxwell AFB, Alabama: Air University Press, 1992),40.

[12] Department of Defense, Data Elements and Data Codes Standardization Program, DOD Directive 5000.11, Since rescinded, was dated December 7, 1964. DOD Directive 8320.1, DOD Data Administration replaced this document when it was published on September 26, 1991.

[13] John G. Roos, In his article "Ending the C4I Tunnel of Babel," states "Decades of independent technical and procurement decisions have left the services with a hodgepodge of command, control, computer, and intelligence (C4I) systems, making it difficult - in some cases impossible - to communicate across service lines. The problem has been obvious for years." Armed Forces Journal, 132 no 3 (October 94): 19.

[14] Peter Rackham, in his article "Uniting the Tower of Babel," states "The US Department of Defense operates the world's most diverse array of information systems with a vast collection of computer architectures, operating systems, hardware, programming languages, software applications, database languages, and protocols. While each information system was designed to do a particular task, many perform similar functions and yet few can interoperate or share data." Jane's Defence Weekly, 22 no 16 (22 October 1994): 26.

[15] Defense Information Systems Agency Memorandum for Program managers and developers, "Implementing DOD Standard Data Elements", (April 1996).

[16] Pam Baylon, <baylonp@ncr.disa.mil>, Information on number of entities in the DOD Data Model and number of data elements in the DDDS provided by electronic mail message subject: "DOD Data Model," to LTC W. Lasher <lasherw@carlisle-emh2.army.mil>, 19 March 1997.

[17] Emmett Paige Jr. (HON), "Achieving the Integrated Systems Concept," Keynote address at the Armed Forces Communications and Electronics Association International Technet '96 Convention, Washington, 4 June 1996, as published in Defense Issues 11 no. 51 (June 1996): 5.

BIBLIOGRAPHY

Andriole, Stephen J. Information System Design Principles for the 90s: Getting It Right. Fairfax Virginia: AFCEA International Press, 1990.

Baylon, Pamela. <baylonp@ncr.disa.mil>. "DOD Data Model." electronic mail message to LTC W. Lasher <lasherw@carlisle-emh2.army.mil>. 19 March 1997.

Bruce, Thomas A. Designing Quality Databases with IDEF1X Information Models New York: Dorset House Publishing, 1991.

Condit, Paul D., MAJ. Principles of Information Resource Management: A Foundation for the Future. Maxwell AFB, Alabama: Air University Press, 1992.

D. Appleton Company, Inc. Process and Data Modeling Workshop using IDEF Techniques Manhattan Beach, California: D. Appleton, 1991.

Martin, James and Leben, Joe. Strategic Information Planning Methodologies Englewood Cliffs, New Jersey: Prentice Hall, 1989.

Owens, William A, ADM "Emerging System of Systems." U.S. Naval Institute Proceedings 121 no 5 (May 1995): 35-39.

Paige, Emmett, HON, Jr. "Achieving the Integrated Systems Concept." Defense Issues 11 no 51 (1996): 1-4.

Rackham, Peter. "Uniting the Tower of Babel." Jane's Defence Weekly 22 no 16 (22 October 1994): 26+.

Roos, John G. "Ending the C4I Tunnel of Babel (Joint Warfare Interoperability Demonstration)." Armed Forces Journal International 132 no 3 (October 1994): 19.

Strassmann, Paul A. "Best Practices? Or Placebos?." Computerworld 30 no 42 (14 October 1996): 97.

U.S. Department of the Army, The Army Data Management and Standards Program. Army Regulation 25-9. Washington: Office of the Secretary of the Army, Director of Information Systems for C4, 25 September 1989.

U.S. Department of the Army, The Army Enterprise Vision. Washington: Office of the Secretary of the Army, Director of Information Systems for C4, 20 July 1993.

U.S. Department of the Army, Army Technical Architecture, Version 4.0. Washington: Office of the Secretary of the Army, Director of Information Systems for C4, 30 January 1996.

U.S. Department of the Army, Army Vision 2010. Washington: Office of the Chief of Staff, Undated, published in 1996.

U.S. Department of Defense, Data Administration Procedures. DOD Manual 8320.1-M. Washington: Defense Information Systems Agency, 29 March 1994.

U.S. Department of Defense, DOD Data Administration. DOD Directive 8320.1. Washington: Department of Defense, Assistant Secretary of Defense for C3I, 26 September 1991.

U.S. Department of Defense, "Implementing DOD Standard Data Elements." Memorandum for Program Managers and Developers. Washington: Defense Information Systems Agency, April 1996.

U.S. Department of Defense, Joint Vision 2010. Washington: Chairman of the Joint Chiefs of Staff, Undated, published in 1996.

U.S. Department of Defense, Life-Cycle Management (LCM) of Automated Information Systems (AISs). Department of Defense, Assistant Secretary of Defense for C3I. Washington: 14 January 1993.

U.S. Department of Defense, Software Development and Documentation. Military Standard 498. Washington: Department of Defense, 5 December 1994.

U.S. Department of Defense, Technical Architecture Framework for Information Management (TAFIM) Volume 1: Overview. Defense Information Systems Agency. Washington: Version 2.0 30 June 1994.