

# Carderock Division Naval Surface Warfare Center

West Bethesda, Maryland 20817-5700

---

**CDNSWC-TSSD-98-005** March 1998

Total Ship Systems Directorate  
Research and Development Report

## **PROPOSED WEB-BASED ARCHITECTURE FOR THE INTEROPERABILITY OF NAVAL AVIATION IETMs.**

by

Eric L. Jorgensen



---

Approved for Public Release - Distribution Unlimited

---

# Report Documentation Page

*Form Approved  
OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

|   |                                    |   |                                  |                                 |
|---|------------------------------------|---|----------------------------------|---------------------------------|
| 1. REPORT DATE<br><b>MAR 1998</b>   | 2. REPORT TYPE                     | 3. DATES COVERED<br><b>00-00-1998 to 00-00-1998</b> |                                  |                                 |
| 4. TITLE AND SUBTITLE<br><b>Proposed Web-Based Architecture for the Interoperability of Naval Aviation IETMs</b>  |                                    | 5a. CONTRACT NUMBER                                 |                                  |                                 |
|   |                                    | 5b. GRANT NUMBER                                    |                                  |                                 |
|   |                                    | 5c. PROGRAM ELEMENT NUMBER                          |                                  |                                 |
| 6. AUTHOR(S)  |                                    | 5d. PROJECT NUMBER                                  |                                  |                                 |
|   |                                    | 5e. TASK NUMBER                                     |                                  |                                 |
|   |                                    | 5f. WORK UNIT NUMBER                                |                                  |                                 |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br><b>Naval Surface Warfare Center, Carderock Division, 9500 MacArthur Blvd, West Bethesda, MD, 20817-5700</b> |                                    | 8. PERFORMING ORGANIZATION REPORT NUMBER            |                                  |                                 |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)   |                                    | 10. SPONSOR/MONITOR'S ACRONYM(S)                    |                                  |                                 |
|   |                                    | 11. SPONSOR/MONITOR'S REPORT NUMBER(S)              |                                  |                                 |
| 12. DISTRIBUTION/AVAILABILITY STATEMENT<br><b>Approved for public release; distribution unlimited</b>   |                                    |   |                                  |                                 |
| 13. SUPPLEMENTARY NOTES   |                                    |   |                                  |                                 |
| 14. ABSTRACT  |                                    |   |                                  |                                 |
| 15. SUBJECT TERMS   |                                    |   |                                  |                                 |
| 16. SECURITY CLASSIFICATION OF:   |                                    |   | 17. LIMITATION OF ABSTRACT       |                                 |
| a. REPORT<br><b>unclassified</b>  | b. ABSTRACT<br><b>unclassified</b> | c. THIS PAGE<br><b>unclassified</b>                 |                                  |                                 |
|   |                                    |   | 18. NUMBER OF PAGES<br><b>39</b> | 19a. NAME OF RESPONSIBLE PERSON |

**PROPOSED WEB-BASED ARCHITECTURE FOR THE  
INTEROPERABILITY OF NAVAL AVIATION IETMS**

by

**Eric L. Jorgensen**

March 1998

Naval Surface Warfare Center  
Carderock Division, Code 2052  
West Bethesda, MD 20817-5700

This Page Intentionally Left Blank

## -ABSTRACT

This report documents the results of a study sponsored by the Naval Air Systems Command (NAVAIR) which is recommending an Architecture for Naval Aviation Interactive Electronic Technical Manuals (IETMs) based on the technology, industry standards, and commercial software products being developed for the World Wide Web

The objective of the NAVAIR Study effort has been to propose a high level IETM Architecture to guide and standardize IETM acquisition, management, and display that:

- will enable, for the end user, maximum interoperability of Technical Information to meet the needs of the Naval Aviation community in supporting the Naval Logistics Information Strategy Plan; and
- will also serve as the basis for a DoD-wide adoption of the proposed approach, to be based on pilot-test programs that will assess the applicability of the Architecture to supporting IETMs for candidate weapon systems of the Military Services.

The recommended Architecture, called the Navy IETM Architecture (NIA), is documented in this report including a summary of what will eventually be four Performance Specifications for the following areas:

- Object-Encapsulation Specification needed for definition of the delivery, transport, and structure of the IETM View Packages.
- Intranet Server and Database Interface Specification.
- Browser Specification.
- Electronic Addressing Specification.

While directed at the requirements of the Naval Aviation Community, the architecture as recommended has been specifically targeted as a Navy wide architecture. It has also been proposed as the basis for a DoD Wide IETM Architecture Study and a formal 18-month effort to conduct this effort has been initiated starting in December 1997 and is sponsored by the DoD CALS Office and the Joint Commanders Group for Communication and Electronics.

## ACKNOWLEDGMENTS

Mr. Jorgensen was assisted in the preparation of this document by the technical contribution and review of members of the Naval Aviation IETM Interoperability Project Team, which he leads. The actual technical input upon which much of this architecture is based will be documented in a separate report. Principal technical-team contributors for the user interoperability portions of the Naval Aviation effort include the following:

NSWC, Carderock Division, Code 2052  
John Junod

ManTech Services Corporation, West Virginia  
Bob Kidwell, Joe Brazy, Don Reynolds, Glen Copen

ANTECH Systems Inc.  
Mike Anderson, Dave Cooper

Funding for the NSWC portions of this effort came from the Naval Air Technical Services Facility, Philadelphia PA. Performing Activity: NSWC/CD Code 2052 Bethesda MD - Eric Jorgensen; E-mail: jorgense@dt.navy.mil

This Page Intentionally Left Blank

# TABLE OF CONTENTS

|   |           |
|---|-----------|
| <b>1. INTRODUCTION.....</b>   | <b>1</b>  |
| 1.1 THE PROBLEM.....  | 1         |
| 1.2 GREATER APPLICABILITY OF STUDY OUTSIDE NAVAL AVIATION.....                                      | 2         |
| 1.3 OBJECTIVE OF STUDY.....   | 2         |
| 1.4 GOAL FOR THE ARCHITECTURE.....  | 3         |
| 1.5 PURPOSE AND SCOPE OF THIS REPORT.....   | 3         |
| 1.6 TECHNICAL APPROACH.....   | 4         |
| <b>2. OVERVIEW OF THE ARCHITECTURE.....</b>   | <b>5</b>  |
| 2.1 IT-21 COMPLIANCE.....   | 5         |
| 2.2 NIA USE OF INTERNET AND WORLD WIDE WEB TECHNOLOGY.....  | 6         |
| 2.3 PROPOSED PERFORMANCE SPECIFICATIONS FOR THE ARCHITECTURE.....                                   | 7         |
| 2.3.1 <i>Object-Encapsulation Specification</i> .....   | 7         |
| 2.3.2 <i>Server and Data-Base Interface Specification</i> .....                                     | 8         |
| 2.3.3 <i>Browser Specification</i> .....  | 8         |
| 2.3.4 <i>Electronic Addressing Specification</i> .....  | 8         |
| 2.3.5 <i>Source-Data Sharing Specification</i> .....  | 9         |
| <b>3. CONCEPT OF OPERATIONS FOR APPLICATION OF ARCHITECTURE.....</b>                                | <b>9</b>  |
| 3.1 NIA OPERATIONAL FLOW DIAGRAM.....   | 9         |
| 3.2 THE USER PERSPECTIVE.....   | 11        |
| 3.3 THE IETM DEVELOPER PERSPECTIVE.....   | 11        |
| 3.4 THE INFRASTRUCTURE PERSPECTIVE.....   | 12        |
| <b>4. ARCHITECTURAL TYPES.....</b>  | <b>12</b> |
| 4.1 CHARACTERISTICS OF ARCHITECTURAL TYPES.....   | 13        |
| 4.2 ELEMENTS DIAGRAMS FOR ARCHITECTURAL TYPES.....  | 15        |
| <b>5. OBJECT ENCAPSULATION FOR DELIVERY, TRANSPORT, AND PRESENTATION OF IETM VIEW PACKAGES.....</b> | <b>18</b> |
| <b>6. EXPECTED PORTABLE ELECTRONIC DATA DEVICE (PEDD) ENVIRONMENT.....</b>                          | <b>18</b> |
| <b>7. BROWSER SPECIFICATION.....</b>  | <b>19</b> |
| <b>8. SERVER AND DATABASE INTERFACE SPECIFICATION.....</b>  | <b>20</b> |
| 8.1 TYPE 1 AND TYPE 2 SERVER SUPPORT.....   | 21        |
| 8.2 TYPE 3 SERVER SUPPORT.....  | 21        |
| 8.3 TYPE 4 SERVER SUPPORT (DATABASE INTERFACE).....   | 22        |
| <b>9. ELECTRONIC ADDRESSING SPECIFICATION.....</b>  | <b>22</b> |
| <b>10. ASSURING AND TESTING COMPLIANCE.....</b>   | <b>24</b> |
| <b>11. MIGRATION AND INTEGRATION OF ELECTRONIC LEGACY SYSTEMS.....</b>                              | <b>24</b> |

|  |           |
|--|-----------|
| <b>12. IMPORTANT UNRESOLVED ISSUES NEEDING FURTHER STUDY .....</b>   | <b>25</b> |
| 12.1 MAINTAINING A COMMON LOOK AND FEEL AMONG DIFFERING IETMS .....  | 25        |
| 12.2 UPDATING VIEW PACKAGES THROUGH THE NAVY INFRASTRUCTURE .....  | 26        |
| 12.3 UPDATING TYPE 4 IETM IMPLEMENTATIONS.....   | 26        |
| 12.4 AUTOMATIC COMPONENT INSTALLATION.....   | 26        |
| 12.5 IMPLICATION OF NON-MICROSOFT IMPLEMENTATIONS.....   | 27        |
| <b>13. BUILDING INTEGRATED PRODUCT SUPPORT DATABASE (IPSDB) USING THE NAVY<br/>IETM ARCHITECTURE .....</b> | <b>27</b> |
| <b>14. ABBREVIATIONS .....</b>   | <b>28</b> |
| <b>15. GLOSSARY.....</b>   | <b>30</b> |

# 1. INTRODUCTION

The transmission of digital data within the Naval Aviation Community, the Navy, and the other Services is quickly becoming the dominant medium for communicating and accessing Technical Information needed to maintain DoD field operations. In response to directives from the Office of the Secretary of Defense, all of the Military Services have ongoing efforts to convert paper-based technical documentation into digital format. They are rapidly replacing existing maintenance and logistic-support Technical Manuals with Interactive Electronic Technical Manuals (IETMs). This data is needed to sustain war-fighting capability in Joint and multi-unit operations, and a uniform approach must be developed for acquiring, managing, and viewing this data. The current practice of independent procurement of Technical Information using divergent technologies and formats must be replaced by a coordinated procedure and guided by an overarching technical architecture which permits IETM applications to work together. Regardless of the source, weapon-support data must be read and viewed by a common user-interface system, and must be accessible from a uniform electronic technical-library interface. Such a common process for managing and deploying digital data will make most effective use of existing resources and will assure maximum interoperability.

## 1.1 The Problem

In 1992 the DoD issued three Military Specifications for Service-wide use in the acquisition of IETMs. These specifications have been successful in their original objective of guiding the development of IETMs, which are now being acquired for many of the DoD's new major weapon systems. However, as individual systems have matured, issues in the area of interoperability between the differing IETM systems, as well as, incompatibility between these IETM systems and the growing inventory of legacy data Electronic Technical Manual (ETM) systems, have arisen. Nearly all early developers of Specification compliant IETM systems had to create both a new authoring system and a user-presentation system as they had no existing products on which to build their development. The net result was that the authoring system and the presentation system developed for individual IETMs were interdependent and an IETM authored by one activity could not be viewed using a presentation system developed by another activity. There was no IETM View Packaging Standard such that a final IETM view Package delivered to the Government would be issued to users who could view the Technical Information accurately with a standard presentation system. Initially, this was not a problem for a weapon-system Acquisition Manager who acquired IETMs, because the developer, typically a prime contractor, was able to control both the IETM and the display system for the dedicated user population for any particular weapon system. But, as the use of IETMs became more widespread, it became important to establish an infrastructure to manage and distribute IETM updates to multiple field sites and to provide life-cycle support for numerous IETMs. In this environment, the fact that differing IETMs cannot interoperate (i.e., cannot be viewed on the same standard presentation system, or electronically reference each other to any meaningful level of internal granularity) has become a major impediment. Additionally, since a common standard for the structure of the

delivered IETM is lacking, it has been very difficult to define the requirements for, or to make the initial design of, a standard infrastructure to support IETMs in the field.

## **1.2 Greater Applicability of Study Outside Naval Aviation**

The Naval Air Systems Command has recognized this problem and, acting in accordance with the Navy Logistics Information Strategy Plan (NLISP) of 8 July 1997, initiated a major study to develop by April 1998 a Navy IETM Architecture (NIA) which will assure wide electronic Technical Information interoperability. This effort is funded by Naval Aviation programs and is being conducted under the technical leadership of the Naval Surface Warfare Center, Carderock Division (NSWC/CD), Code 2052, Bethesda MD. When this study is completed, it will be presented to NAVAIR Policy Officials who will make any decisions as to the extent that the recommendations contained in this report will become NAVAIR Policy or required practice.

Additionally, the DoD Tri-Service IETM Technology Working Group (IETMTWG), chartered by the OSD CALS Office of DUSD(L), has endorsed the NAVAIR Project Plan as an approach which offers the potential for a DoD-wide solution. At the request of the OSD CALS Office, the IETMTWG has proposed to expand the NAVAIR project into a DoD-wide effort that involves prototyping and testing the NAVAIR improved interoperability methodology using a Tri-Service spectrum of weapon systems. The proposed IETMTWG plan has been reviewed by the Technical Publications Sub-panel of the Joint Commanders Group for Communications and Electronics (JCG-CE) as a solution to one of the major goals of the JCG-CE Publications Panel, the achievement of field interoperability for IETMs. The proposed approach was approved and the JLC has recommended, by a letter of 10 June 1997, that the OSD CALS Office implement this plan as a joint effort between the JCG-CE and the IETMTWG. The actual effort to initiate this DoD wide effort technically started in late 1997 and will continue through June 1999. NSWC/CD will also lead this DoD effort.

## **1.3 Objective of Study**

The objective of the NAVAIR Study effort has been to create a high level IETM Architecture to guide and standardize IETM acquisition, management, and display that:

- (1) will enable, for the end user, maximum interoperability of Technical Information to meet the needs of the Naval Aviation community in supporting the Naval Logistics Information Strategy Plan;
- (2) will also serve as the basis for a DoD-wide adoption of the proposed approach, to be based on pilot-test programs that will assess the applicability of the Architecture to supporting IETMs for candidate weapon systems of the Military Services.

## 1.4 Goal for the Architecture

The primary goal for the Navy IETM Architecture (NIA) is to set the foundation for the acquisition and deployment of sharable and interoperable technical data is distributed to the work location of an end-user, who will be able to view and utilize that data through a common user interface, no matter what the authoring source or data format. In so doing the Naval Aviation Community will be able to establish a unified approach to the acquisition, management, and use of existing ETMs and newly procured IETMs. To meet this goal, the overall approach will be based on the use of existing COTS and NDI Internet and World Wide Web technology. An overall end goal is to achieve end-user-level interoperability of the IETMs delivered to and used by the Naval Aviation Community. In this context, an IETM is defined as having end-user interoperability when it can enable a user with a common, commercially available display device, such as a portable personal computer:

- (1) to view and interact with an IETM from any source and of any internal format; and
- (2) to view, by means of an electronic-link reference in the displayed IETM, information in any other IETM to which the link refers.

## 1.5 Purpose and Scope of this Report

The purpose of this report is to describe the portions of the Navy IETM Architecture applicable to end-user interoperability so that three major constituencies can develop needed capabilities for a Naval Aviation Interoperable IETM End-User Capability. The three targeted constituencies are:

- (1) The creators of the IETM products (both content providers and presentation-software vendors);
- (2) The developers of the IETM user-infrastructure (in the case of Naval Aviation this will be a part of the planned Automated Maintenance Environment - AME); and
- (3) The procurers of the common display devices together with the common browser software installed on the devices.

The report is also intended for NAVAIR Logistics Managers and Acquisition Program Managers who are responsible for policy and direction of these constituencies. This report is thus intentionally a focused technical description and not primarily a tutorial, although some of the explanatory material included herein is instructive in nature.

The NIA has been developed to provide interoperability for all levels of Electronic Technical Manuals including all five ETM/IETM Classes from the digitized page-oriented TMs to the highly Interactive Electronic Technical Manuals. For purposes of this report, the term

“IETM” will hereafter be used to refer to all classes of ETM/IETMs whether the class definitions call them ETMs or IETMs.

The important area of interoperability of the source data used to prepare the IETM View packages will be presented in another report and is only summarized herein for purposes of overall perspective and to present the complete vision for the Architecture.

In addition to the Naval Aviation focus of the architecture, this report is also intended to be a baseline description of a DoD wide Architecture and is expected to serve as the basis for expansion of the effort into a Tri-Service project in accordance with IETMTWG / JCG-CE plans.

## 1.6 Technical Approach

The overall concept of this effort is to utilize the group of emerging technologies that the commercial marketplace is rapidly adopting as the standard for electronic documents based on the technology of the Internet and the World Wide Web. For security and operational reasons, the Navy will not, of course, utilize the actual Internet or the World Wide Web itself, but can employ essentially the same technology and COTS products in a private and dedicated DoD intranet environment. Such an approach is becoming the de facto standard for corporate information-distribution systems worldwide. Once this approach has been proved effective, a set of implementation standards will be developed within this comprehensive, DoD-wide, commercially supported (i.e., COTS) framework.

A major objective of the effort to develop the Architecture is to demonstrate end-user interoperability of proprietary and legacy Electronic Technical Manuals by encapsulating them into a common IETM View Package (VP) format which can be viewed by the end user employing a single commercially available user information interface, a process referred to in this report as "object encapsulation". This demonstration requires the establishment of the following technical capabilities:

- (1) an authoring system to effectively create and manage IETMs (regardless of which authoring tool, etc., is used);
- (2) an infrastructure that permits a military component to distribute, manage, and present these IETMs; and
- (3) a system that permits an end user to perform his job effectively through access to required Technical Information, and that allows him to retrieve relevant data from other IETMs, including those of other Services, if necessary.

In order to achieve interoperability, the performance specification recommended for the NIA will be specific, but with the clear intent to not preclude innovative solutions, especially in light of the constantly expanding technology base. Achieving this balance has required

making some decisions that may need to be reexamined over time. Whenever possible, the design adheres to open standards or de facto standards widely implemented by multiple vendors.

## **2. OVERVIEW OF THE ARCHITECTURE**

The NIA (Navy IETM Architecture) is firmly based on the proven and widely accepted Internet and World Wide Web technology, implemented as a private Web on a contained intranet. This intranet can be configured as a private DoD World-wide network (e.g., the Global Combat Support System – GCSS), as a ship or squadron-wide network (e.g., the NAVAIR AME network envisioned for all Naval Aviation sites), or as a group of computers in close proximity hard-wired in an Ethernet configuration. It can also be configured in a single display device (portable or workstation personal computer) which operates both as a client browser and a personal single-user Web server. The technology for implementing such an intranet is low-risk, easily implemented, and widely understood. The proposed architecture is based entirely on COTS and NDI technology. The architecture is based on a dedicated Web or intranet that, at a minimum, has at least one Web-browser client, at least one Web server (more precisely, an HTTP server and its included file-based store), and a network to connect them. The specific implementation of the network, which is typically a TCP/IP based network when more than one device is involved, is not discussed in detail in this report and will typically vary from one implementation to another. As will be described more fully below, the intranet may include optional database servers and application servers as well as the HTTP server.

### **2.1 IT-21 Compliance**

The Navy IETM Architecture will be compliant with the Navy's Information Technology for the 21<sup>st</sup> Century (IT-21) initiative, which standardizes the operating environment by employing the Microsoft Windows NT Workstation and the Windows NT Server across the entire suite of non-weapon-system-specific computers and applications. Microsoft NT technology also includes networking capability and automatically includes many Internet/Web oriented services in the NT operating system

The recommendations developed by this Effort are specifically directed towards the Naval Aviation Community and are intended to assure operation in the Navy IT-21 environment. In this light, it is recognized that in certain cases, especially those involving advanced IETMs that require application and database services or the brokering of distributed components and no single accepted de facto open standard exists, the Microsoft implementation of many of these technologies will be recommended. The basic Architecture, however, is not intended nor constrained to be a Microsoft specific Architecture, and can be adapted to non-Microsoft implementations for other DoD application.

The breadboards for this project were developed in a Windows NT environment using both Microsoft NT Server and NT Workstation products. In fact, if the Naval Aviation Community were fully operational with IT-21 hardware and software, the planned

information architecture would be very much easier to establish, as well as much less risky to implement. Because of this de facto standardization and the benefits it will offer for IETM interoperability, the adoption of IT-21, as it relates to IETM storage and display devices, is strongly supported for the Naval Aviation Community.

## **2.2 NIA Use of Internet and World Wide Web Technology**

The approach to developing a solution for this interoperability problem has been to adapt commercial and industry applications involving electronic documentation for which there is widespread vendor-product support. The Naval Aviation IETM Information Architecture Project is applying the products and standards being developed for the World Wide Web and the Internet in a dedicated private-intranet environment. The NIA has intentionally been designed to be extensible, flexible, and able to accommodate the predictable rapid growth in technology for all aspects of the Internet, the Web, and the emerging electronic documentation applications being developed to operate on the Web. The Web is, by its nature a client/server architecture and there is one area on the client/server spectrum in which NIA compliant IETM Applications may differ in emphasis from a major server-centric trend that is emerging for many commercial “enterprise” applications. The NIA is intentionally biased towards a client-centric model employing encapsulated objects that are downloaded to a portable device for use. The server is treated as a utility electronic bookshelf with the IETM View Packages (i.e., the encapsulated objects) designed so they can easily be moved to another electronic bookshelf at another physical site, reflecting the operational reality of the military unit itself. On the other hand, commercial Web sites tend to be permanently located corporate resource centers at which both the servers and the information providers are located. For these commercial activities, the mobile and less controlled entity is the user client. In this scenario, the preference is towards server-centric computing and the use of server-oriented Web-object components. The corporate personnel resources for maintaining both the Web server and the content are located at the Web site. In the military, the server sites have more of the characteristics of a technical library and not a computer information center. The content related technical expertise lies with the content creator or the end user. This situation at this time favors total object encapsulation and client-centric computing as the primary emphasis of the NIA.

Progress in Web-oriented technology and the state of the availability of secure and affordable military global intranets may well change this situation in the future. Thus, the NIA proposed below is intentionally designed so to not preclude such server-centric solutions should such a change occur. In this light, it is important to emphasize that any implementing policy for the NIA must include some specific guidance on how to apply the Architecture, as well as, the requirement to conform to the architecture. The use of custom servers is an important for which such guidance must be matured over time. Guidance documents for the NIA and any possible DoD wide expansion of the NIA must be continually updated over time. Such updates must be based on a continuing study of the emerging Military requirement compared to the current state of commercial technology and available COTS commercial products. The Naval Aviation Community or any other DoD component can not

simply buy the latest and greatest commercially available technology without checking it against real Military requirements, which are not always the same as commercial requirements for which the products are often created

Following is a summary of initial recommendations for the Naval Aviation implementation of the NIA, as well as, the baseline requirement for the NIA.

## **2.3 Proposed Performance Specifications for the Architecture**

In addition to assuming the widely known and accepted Internet/Web standards utilized in building any intranet based on the International W3 Consortium Standards, the IETM Improved Interoperability Architecture is being specified in the form of performance and interface specifications in the following areas:

- Object-Encapsulation Specification needed for definition of the delivery, transport, and structure of the IETM View Packages.
- Intranet Server and Database Interface Specification.
- Browser Specification.
- Electronic Addressing Specification.
- Source-Data Sharing Specification (to be documented in another report)

The Object-Encapsulation, Intranet Server and Database Interface, Browser, and Electronic Addressing performance specifications are required to effect interoperability of disparate IETMs in the field. Achievement of interoperability implies the ability to view any IETM with any browser that conforms to the IETM Browser Specification. It thus requires that all cross references by one IETM to another IETM be encoded in a manner such that the IETM browser will be able to access the referenced IETM by a simple selection button "push" (e.g., mouse click). In addition to these end-user interoperability specifications, the eventual complete Architecture recommended for Naval Aviation will include a Source-Data Sharing Specification in order to achieve interoperability of source data; that is, the ability for one authoring environment to automatically import source data from another authoring environment. The details in this Source-Data-Sharing Specification will be established by an additional phase of the study, which is still ongoing at the time of this writing. Below is a short summary of the five specifications with a more detailed discussion of the first four presented later in this report.

### **2.3.1 Object-Encapsulation Specification**

A core philosophy underlying this architecture is that developers of IETMs can package and deliver, as a single data package composed of encapsulated objects called a View Package, all capability and content for an IETM that is needed to use the IETM on an unmodified standard Naval Aviation Intranet. This View Package may in fact contain both content data and software components and can be treated as an encapsulated data set for purposes of

contract delivery to an electronic archive or subsequent store-and-forward management site. It will eventually be delivered by the Naval Aviation Infrastructure to the Fleet user activities as though it were a simple data package. Similarly, it will be treated by the Infrastructure as file-oriented data for the User Intranet Web Server, i.e., simply as a generic "bucket of sequenced bits" which makes sense to the server but is of no concern to the infrastructure as long as it is kept together. Essentially the View Package is a set of industry standard binary files, each of which is assigned a notional URL which contains sufficient information for installation as data in the Intranet Server file system. Until the point of receipt by the intranet server, the View Package is processed as a single object. There are a variety of mature approaches for bundling a set of files with headers into a single data set (e.g., INTERNET MIME Standards) and the Architecture may use any of them, requiring only that the View Package can be installed as a set of files on the intranet server. With this approach, no overt man-in-the-loop software installation processes are required other than the automatic capability built into the World-Wide-Web-capable browsers and servers.

### **2.3.2 *Server and Data-Base Interface Specification***

The simplest way for the NIA to achieve IETM interoperability for the Naval Aviation Community is to utilize only primary generic servers with widely available server extensions such as the Microsoft Front Page and Active Server Page extensions. Such an approach will require no additional software to be overtly installed on either the servers or the browser device. However, it is recognized that some legacy systems, and possibly some highly innovative new IETM applications, may require some sort of custom server extensions and database interface components. Final recommendations on the use and encapsulation of server extensions will require additional technical investigations as the technology and marketplace needs to mature before a full tradeoff and the development of specific recommendations can be accomplished.

### **2.3.3 *Browser Specification***

The Browser Specification will specify the versions of the two dominant commercial browser products and a set of standard extensions (i.e., controls and/or plug-ins) to these browsers, which include common DoD data viewers such as PDF, a SGML viewer, CGM Version 4 Graphics, and CALS raster images. The utility, functionality, maturity, and IT-21 compatibility of the DCOM family of object broker standards is such that it will be recommended for the Naval Aviation implementation of the NIA. While Internet Explorer fully supports DCOM, there is a need for an extension of the Netscape browser to process Active-X Controls, the needed IETM related aspect of DCOM. The eventual goal is to have all valid DoD IETMs be compatible with both the Internet Explorer and Netscape products, possibly requiring some installed extensions.

### **2.3.4 *Electronic Addressing Specification***

The Electronic Addressing Specification will be based on the existing Universal Resource Locator (URL) standard for the World Wide Web because it is widely implemented in virtually all Web-enabled vendor products. Any occurrence of a legitimate URL string of characters is automatically made "hot" in the vendor application and a mouse click or two on

that hot spot will launch a Web browser search which in turn will locate the file referenced by the URL and display it on the screen. In addition to requiring a standard syntax, the Electronic Addressing Specification will also require that all of the Services maintain and publish a permanent registry of all valid references to the IETMs issued by that Service. Once published, a valid URL must not be changed. This type of URL is called a Persistent URL (P/URL). The specification will, of course, address the requirement in an intranet environment to allow the remapping of these P/URLs (which reference a hypothetical server on the World-Wide Web) into the actual server and file-system locations on the intranet under use.

### **2.3.5** *Source-Data Sharing Specification*

A fifth specification, the Source-Data Sharing Specification, is not documented in this report and is being separately developed. This Specification, which seeks to achieve a goal that has been sought for almost ten years, will be slow to be fully implemented and will be dependent on achieving a consensus of the developers of the major IETM-authoring approaches in the DoD and its supplier base. As such, its final form has not yet been formulated, but will be based on a standard SGML-specified common denominator that reflects the architectural forms in MIL-PRF-87269, the IETM database specification. However, Source Data interoperability is not necessary to achieve the specific goals of the end-user interoperability objective presented in this report.

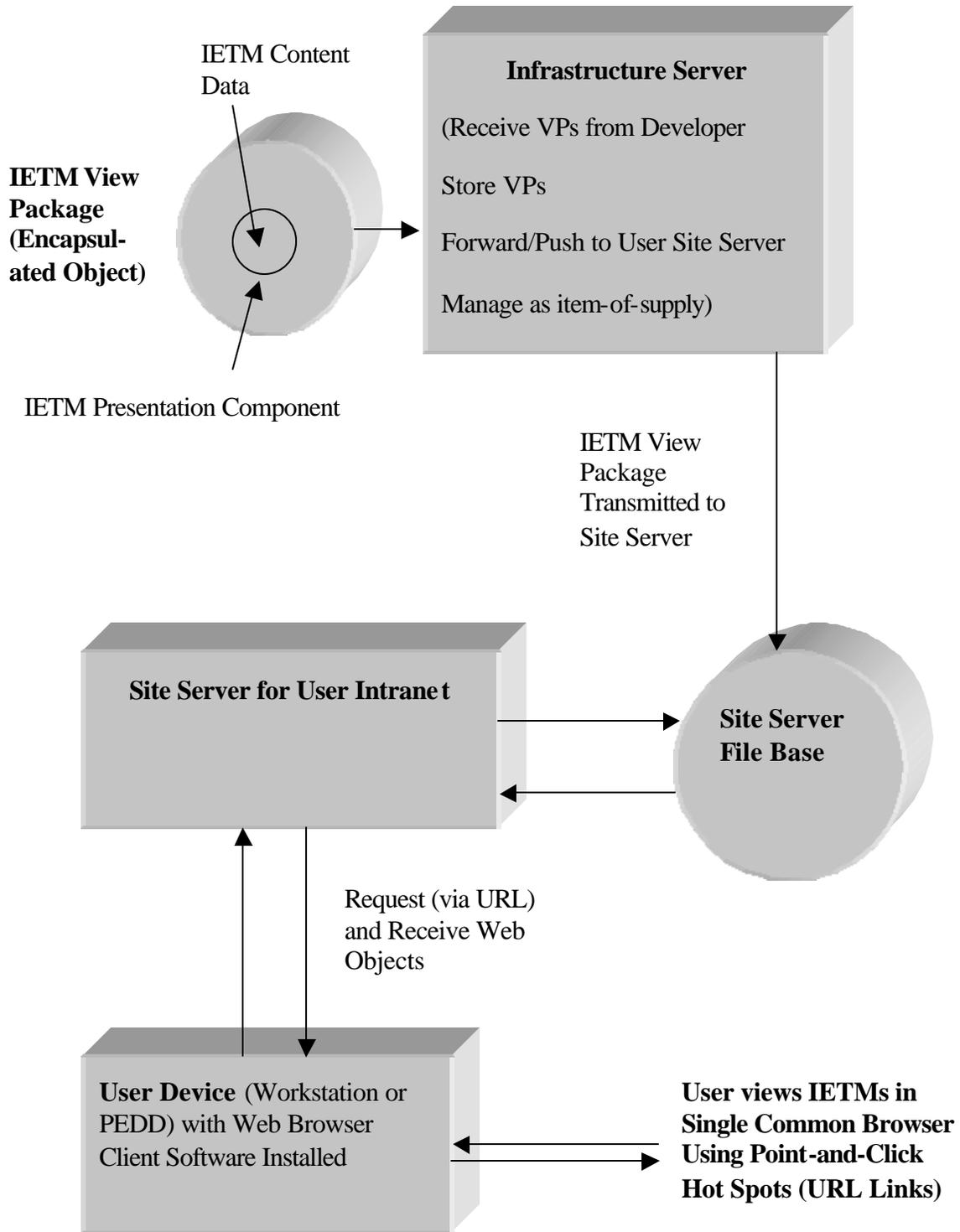
When Source-Data interoperability is achieved, it will operate very well with in the end-user Architecture. To encourage the use of the Source-Data Specification for the exchange of sharable information among various suppliers of IETMs, it is expected that the requirements for a standard viewer for this data structure will be specified to conform fully to the recommended Browser Specification. This will allow easy object encapsulation of a user-viewable version of the sharable data structure.

## **3. CONCEPT OF OPERATIONS FOR APPLICATION OF ARCHITECTURE**

### **3.1 NIA Operational Flow Diagram**

The following (Figure 1) illustrates the flow of an NIA implementation from the original IETM developer, through the management infrastructure repository, to the user-site intranet server, to the Web browser viewing area and eventually to the user who selects the next object to view via a point-and-click Web-browser interface. The presentation components referred to can be client (Type 1) or server (Type 3) components or implied (i.e., omitted) in the case when they are preinstalled in the standard browser (Type 2). These Architectural Types are variants of this overall flow diagram and are described in Sections 4 and 8 below.

**Figure 1 – Flow of IETMs through the NIA**



## 3.2 The User Perspective

The end user accesses and views the IETMs in the following manner. The typical device that a technician will access is a workstation personal computer or a PEDD (Portable Electronic Display Device). This device will be configured either as a network client attached to the squadron intranet or it will be configured to operate in stand-alone mode. In either case the man-machine interface is identical and the user cannot determine from the look and feel of a display in which mode the device is operating. To access an IETM, the user will employ an URL reference using one of the many access-screen or menu-select options available (e.g., favorites list, explicit entry, a pre-assembled list of active IETMs on a squadron Home Page, a hot-spotted index graphic, a Web-page job-assignment form listing the needed technical references). All of these are common practices borrowed from the World Wide Web community. From the user's perspective the referenced IETM simply appears in the browser window. Depending on the browser security level set, the user may at times need to overtly accept components that require installation, but no other explicit installation action is needed as the browser installs the components automatically. This is a key user-friendly feature of the NIA. Thus, there is no need for a system administrator to install user software; that is a part of the simplicity of this approach. Web access is a proven "point and click" user interface. If one IETM contains a reference to another IETM, the user can click on the reference and that IETM will appear in the browser window (assuming, of course, it is installed on the user's intranet). This second IETM can in turn reference a third IETM. To return to the original IETM, the user simply uses the "back" arrow on the browser interface, effectively reversing the references. Modern Web browsers can handle many levels of such nested referencing with no performance degradation, a very powerful feature. From the user perspective the NIA is thus intended to make the use of the disparate IETMs as easy and "seamless" as possible with modern technology.

## 3.3 The IETM Developer Perspective

The principal emphasis from the IETM-developer perspective is that all software components and data needed to make an IETM accessible on the NIA display device are bundled into a single data product (i.e., the encapsulated object), which is easily installed as a set of data files onto an intranet-server file system. This set of encapsulated objects is called a View Package. All data and component delivery to the end user is accomplished through the Web-based client-server interaction. An additional feature is that this View Package can be passed, unmodified, from server to server as part of the NIA electronic-distribution system. While the technology needed to accomplish this transfer is complex, it is off-the-shelf and neither expensive nor difficult to obtain. This is due to the exploding popularity of the Internet and the World Wide Web for commercial applications and the rush by suppliers to get competitive products to market. A foundational principal of the NIA is that the products developed for the Internet can be used unmodified to develop IETM products for an NIA-compliant Intranet. This process is in sharp contrast to a conventional IETM application where the IETM product is

delivered as two separate items, the IETM content data and the IETM presentation-system application-software program. The later requires an explicit installation process onto every applicable end-user device even if it is co-located on the same CD-ROM as the data. There are many options for the required object encapsulation, however, those specific methodologies are not specified in the NIA. Only the Web-based interface to the object is specified.

### **3.4 The Infrastructure Perspective**

The key NIA concept for the Infrastructure designer and user is that the IETM View Packages are composed of self-contained digital objects which appear to the infrastructure as large standard binary formatted digital files. These objects can be received from a developer, stored, forwarded, and delivered from one server to another without the end user's ever needing to know the internal structure of the View Package itself. The infrastructure site can function more as a supply center than as an information-systems center. The specific design of and development of recommendations for a Naval Aviation Infrastructure was not in the scope of this reported effort. This will undoubtedly be a complex, difficult, but important task that will be complicated by the impact it will have on many existing business practices. However, this key NIA element, that the objects can be processed as an item of supply, with no requirement to manage the internal content or structure of the object, should make this task much more manageable.

## **4. ARCHITECTURAL TYPES**

The following breakdown of anticipated IETM View Packages by Architectural Type is presented at this point in this report; it will be described in more detail in subsequent sections. Definitions of these Architectural Types are given in Table 1. These Type definitions group into two areas:

- (1) The core architecture, which applies to IETM Architectural Types 1 and 2. Definition of these two Architectural Types has essentially been completed. These client-centric (sometimes called "fat client") Types require only a browser and a generic HTTP server.
- (2) The extended architecture, Types 3 and 4. For these server-centric Types, the technology for employing the additional servers in the Web environment is less mature and more diverse. It is just now emerging, and is still dominated by proprietary products. (This situation is in large part due to the fact that vendors have opened the browser products to the public domain, a market in which there is little money to be made, and have kept the proprietary-server market, where they see profits to be obtained and seek competitive advantage.)

## 4.1 Characteristics of Architectural Types

Types 1 and 2 share common important characteristics in that they do not require any unique software to be installed on, or to operate on, the server. Thus, the server can be treated as an electronic bookshelf. As far as the server is concerned, the two parts of an encapsulated object (the data and the associated software components) are simply treated as files on the server. Additionally, any contemporary HTTP server can be employed and it does not matter what operating system is employed. Thus, for Type-1 and Type-2 IETM applications, interoperability is very low-risk in the sense that, with these, any IETM View Package can be accessed using any server. For the server requirement for Types 1 and 2, only a generic server is required and no NIA specific server specification is required. Both Types 1 and 2 are considered pure encapsulated-object Types, however, for Type 2, the component part of the object can be implied, i.e., omitted, as it can be assumed to be preinstalled on any NIA compliant browser and need not be included in the transported IETM View Package.

Type 2 is closely tied to the definition of “HTML/XML”, which needs further clarification. For planning purposes it is recommended that foreseeable emerging standards (and not current standards) be used to define the NIA requirement. In this light, HTML/XML is herein specified as employing both HTML version 4.0 and XML version 1.0 (including the associated XSL style and XLL linking specifications), when these two International W3 standards are formally approved. HTML 4.0 is mature and near approval, while, the XML family of standards is still a year or two away. There are several reasons for this recommendation. The future standards will almost certainly be relevant in the time frame when most applications are developed according to the proposed architecture, so the best estimate as to what will be applicable should be used. Another important consideration is that there is written commitment of essentially all the major software vendors to support the future standards, whereas there is no complete agreement on the delivered-product support of the current standard (i.e., HTML 3.2). In particular, vendors have indicated support of the emerging HTML 4.0 standard. Additionally the XML standard has also enjoyed widespread vendor promise of support. XML lags behind HTML 4.0 in maturity, but is essentially complete, and promises to be the user-definable expansion of HTML, and one that is more suited to complex IETMs than HTML. In particular, it will be much easier to convert the large Navy inventory of SGML-tagged source data to XML for a run-time object than it is to convert it to HTML for presentation.

**Table 1 - Proposed IETM Architectural Types**

| Type                               | Characteristics  | Examples   |
|------------------------------------|--|--|
| <b>1a<br/>Simple<br/>Component</b> | One data set plus one custom automatically downloadable non-HTML component | .pdf plus Acrobat reader control<br>.doc plus WordView control |

| Type  | Characteristics   | Examples  |
|---|---|---|
| <b>1b<br/>Complex<br/>Component</b>               | Nested Type 1 data-set/component pairs (“encapsulated objects”). First component loaded into browser shell/container has capability to access another client component and associated data set under control of original component. Requires component licensing. Not recommended for new applications. Client processing only. Uses “plain vanilla” HTTP server.   | Legacy Systems reprogrammed as custom browser or presentation system operating inside a standard browser shell/container. |
| <b>2a<br/>Basic<br/>HTML<br/>Pages</b>            | HTML/ XML page(s) with only browser-resident components. Requires no component licensing. Most will work on any browser. Client processing only. “Plain vanilla” HTTP server.   | HTML with Java script, GIF, JPEG, frames  |
| <b>2b<br/>Compound<br/>HTML<br/>Pages</b>         | HTML/ XML page(s) plus one or more custom components of Type 1. May require component licensing for custom components. Client processing only. “Plain vanilla” HTTP server.   | HTML file plus Java bean(s)<br>HTML file plus plug-in<br>HTML file plus ActiveX control(s)                                |
| <b>3<br/>HTML Plus<br/>Application<br/>Server</b> | Two-tier architecture in which Web page includes reference to server application(s), which must operate before page, is delivered to client as Type 2 HTML/ XML. Data and components managed on server). May utilize database co-located on server but most content is in web page files.<br><br>Requires HTTP server with components for server-side computations. Requires client <i>and</i> server processing.               | MS Front Page Webs<br>MS Design-time Controls<br><br>CGI Server Apps<br>DynaWeb   |
| <b>4<br/>HTML with<br/>Database<br/>Server</b>    | Three-tier architecture that includes a Web page server with pages functioning like a template; e.g., for calls to a database, which contains most of the IETM content. Can include server and components for custom functions. Requires a database server (e.g., Oracle) in addition to the HTTP server. Can use MIL-PRF-87269 Data model for data base on DB server.<br><br>Permits both Client <i>and</i> Server processing. | AIMSS 4.0 (planned)<br>GD TechSight Web<br>MS ASP w/ODBC Calls  |

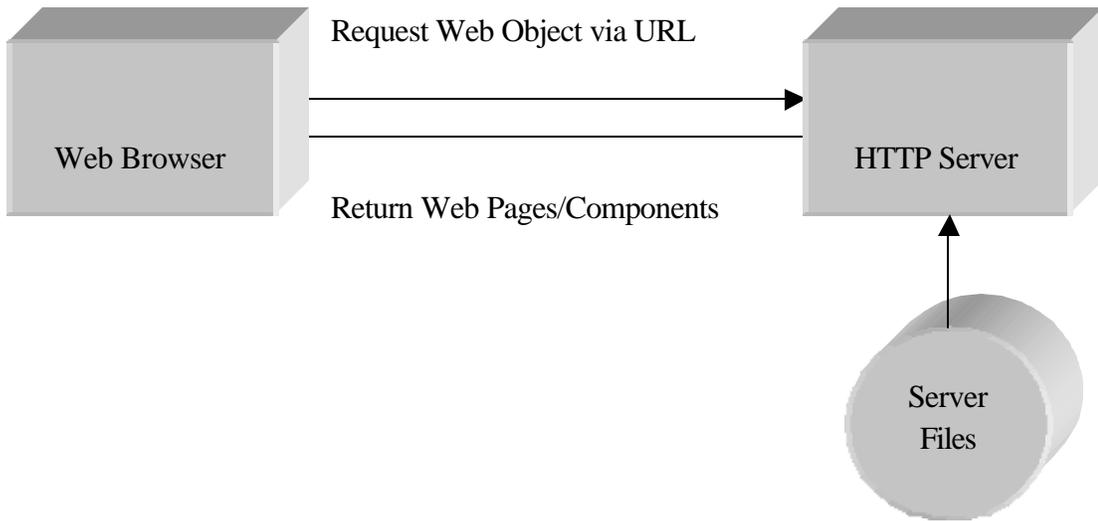
For Type-3 and Type-4 IETM applications, particularly for the server application (i.e., software), the situation for ascertaining de facto industry practices is much more complex. Several approaches are available for standardizing many of the issues such as Microsoft’s design-time controls, Active Server Pages (ASP), and Front Page server extensions, and a variety of third-party middle-ware products; but they are all proprietary and not universally accepted. The technology and state-of-COTS are not sufficiently mature at this time to propose any one of them as a DoD standard so that all IETMs can operate on a single server. To achieve operational interoperability for Types 3 and 4 in the short term, there are two possible approaches for a working solution:

- (1) The various IETM providers must put their own physical server(s) plus the IETM View Packages on the shared user intranet (very feasible with the state-of-the-art and capacity of today's portable computers and plug-in network standards); or
- (2) Require that all IETM creators use the same sets of server components and that the standard components be installed on all intranets employed in the community throughout which the IETMs are interoperable.

## **4.2 Elements Diagrams for Architectural Types**

The core Architecture requires at least two kinds of software elements: one or more client browsers and one or more Web servers, as illustrated in Fig. 2. In general these are hosted on separate devices connected by a TCP/IP network (i.e., LAN); however, an intranet can be set up in a single display device without a network. In the case of IETM Architectural Types 1 and 2, these two kinds of elements are all that is needed. In the case of Type 3 (see Fig. 3), there is a requirement for an additional element, the application server, sometimes referred to as a Web-server extension, since it effectively operates in the same operating system as, and is an extension of, the HTTP server. In the case of Architectural Type 4 (see Fig. 4), there is the additional requirement for a Database Server which hosts most of the IETM content, which may or may not be hosted in the same device as the Web server. Type 4 is also a Type 3, as it requires an application server to process the data-access request dialog between the Web server and the separate database server. Note that the line between Types 3 and 4 may, at times, not be clear as is the case where the application server performs some data base functions; but in general they differ in where the primary data content is stored – server files or database server.

**Figure 2 - Elements for Architectural Types 1 and 2**



**Figure 3 - Elements for Architectural Type 3**

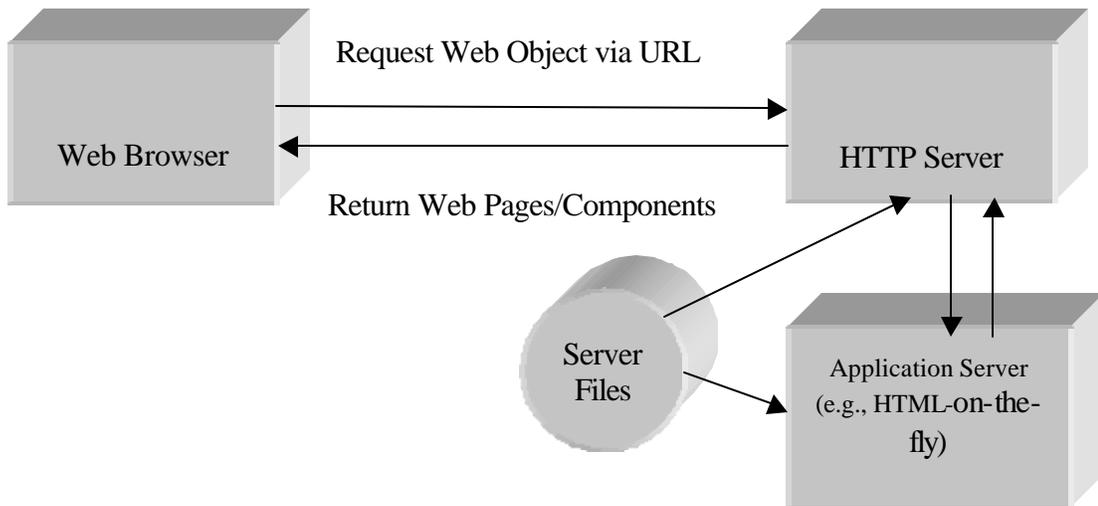
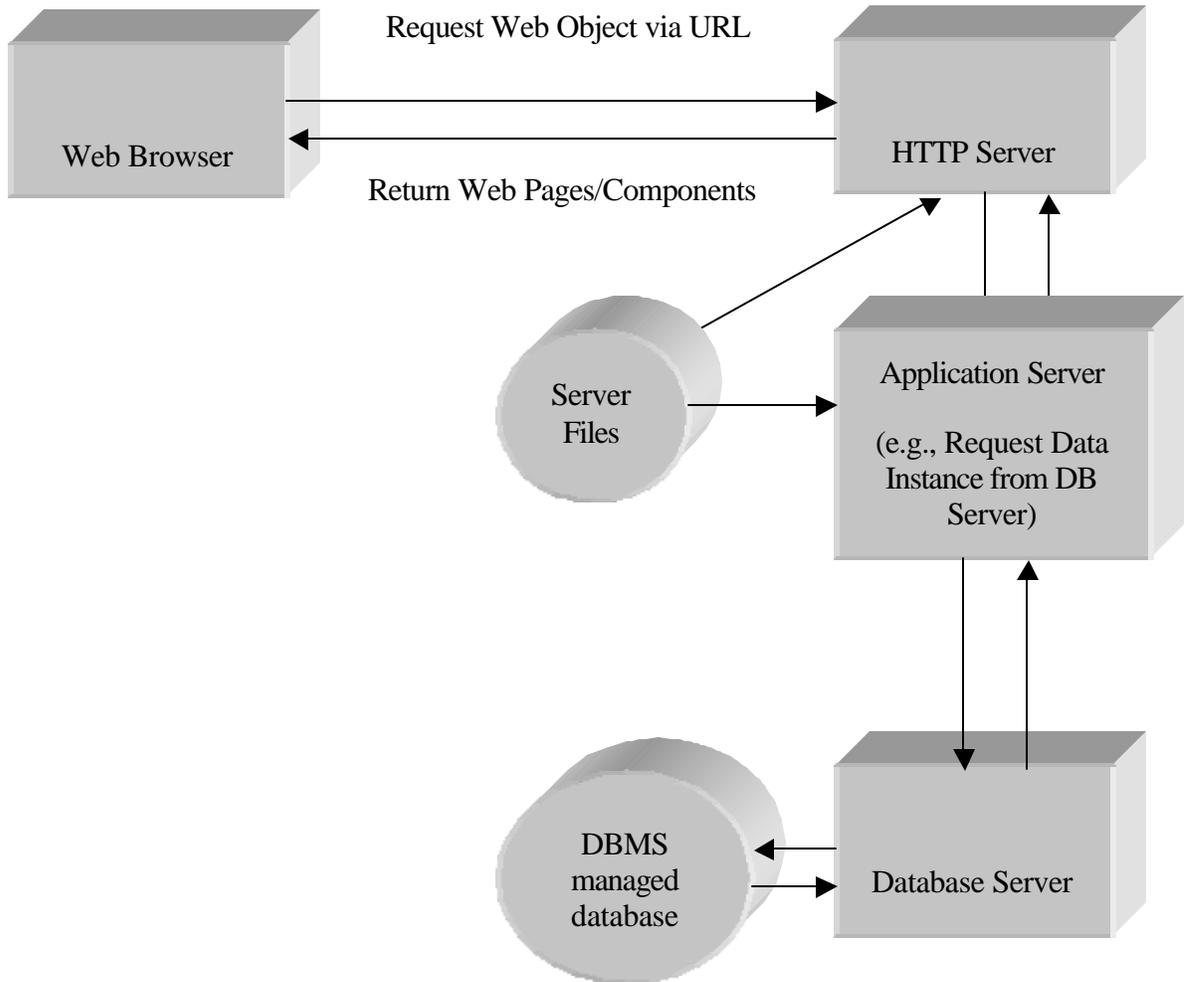


Figure 4 - Elements for Architectural Type 4



## **5. OBJECT ENCAPSULATION FOR DELIVERY, TRANSPORT, AND PRESENTATION OF IETM VIEW PACKAGES**

Of the four proposed specification areas, the Object-Encapsulation Specification, leading to construction of the IETM View Package, will allow the most variability within its performance requirement. The other specifications are more detailed. Only the interface to the object to the Web servers and browser is specified in the NIA Object-Encapsulation Specification, not the internal entities within the object. The basic concept of the Object-Encapsulation Specification is to ensure that all of the data for a particular IETM, and all of the methods or processing instructions for the viewing of the data, are included in individual data packages or distributable units called encapsulated objects. The IETM View Package is thus composed of a combination of linked, encapsulated objects, a digital product that is delivered to the Government as the end-user version of the IETM. The Specification will require that encapsulated objects can be automatically loaded onto an intranet server and that the server can provide those objects to the IETM browser on request. To be effective, the Specification must also specify or reference the capabilities of the intended server and will need to coordinate to the browser specification so that the View Package software components can be automatically loaded on the browser and server as required. For the NIA to be compatible with the Navy IT-21 initiative, it must support Microsoft DCOM object brokering standards, especially those relating to Active-X controls. The inherent capabilities of the NIA compliant browser will include basic presentation methods, either native to the commercial browser or added to meet NIA requirement, so that the component portion of the encapsulated object can be assumed to be preinstalled on the user device. In most cases these particular components need not be included in the View Package. Native browser support includes components such as HTML layout, GIF viewers, and JPEG display. Anticipated NIA specified components may include components such as PDF viewers and Version 4 CGM viewers.

## **6. EXPECTED PORTABLE ELECTRONIC DATA DEVICE (PEDD) ENVIRONMENT**

A unique feature of the Naval Aviation Intranet, as opposed to more conventional intranets, is that the common mode for the PEDD (or other display device) to operate is as a stand-alone, unconnected to any network when the work is actually being performed.

The NSWC/CD laboratory has shown that it is possible to bring all the functionality of a distributed intranet to a single device by installing a personal Web server on the PEDD and, to the extent needed, all other servers which might be needed for Architectural Types 3 and 4. This is not difficult to do in practice; with Microsoft NT Workstation, the Web server is automatically included in the operating system as "peer Web services". For Architectural Type 4 IETM applications, when the PEDD is used in stand-alone

mode, there will be a need to explicitly install the database management system (DBMS) which performs the database-server function. The Web server is included in the NT Workstation operating system as “peer Web Services”, which merely needs to be activated in the operating system. No separate Web server is required.

There is also the need to perform server-request redirection by which all server requests in URLs are redirected back onto the PEDD server file system. There are several off-the-shelf approaches to accomplish this function (e.g., Windows HOST file, Local DNS, etc.) and all will implement the architecture. As actual applications are developed, there will be, of course, a substantial requirement for configuration management facilities to be built into the downloading system that is supplying data to the PEDDs. However, with these self contained intranet features in place, all of which are standard parts of the NT operating environment, it is possible to access any object loaded onto the device in exactly the same fashion as from the site server.

## **7. BROWSER SPECIFICATION**

In line with the COTS and Industry Standards philosophy of this Architecture, the browser requirements are basically established by two particular commercial products, which together have essentially captured the entire Web-browser market. While it is possible to develop, assess, and evaluate a long list of needed and desirable requirements for the IETM browser, such an exercise would serve little purpose in light of economic and market place realities. On one hand new Web browsers are software products that are very complex and expensive to develop; on the other hand, the current products are being offered in the market place free of charge, effectively precluding the development of additional commercial general-purpose browser products. At this writing, these two products are Netscape Navigator version 4 and Microsoft Internet Explorer version 4. Except for a few, but very important capabilities discussed below, these two products are functionally identical. For the traditional HTML 3.2 and earlier Web pages that dominate the WWW, they perform in a similar fashion.

One major area of difference is in the area of object brokering and automatically downloadable components. Ideally, it would be desirable to require that IETMs operate with either browser; however, the Naval Aviation Study team has concluded that such a policy would restrict a very needed capability. Regarding downloadable and automatically installable software components, the two products differ in a marked degree. This is largely due to Netscape’s overt unwillingness to support Microsoft distributed object-broker standards (i.e., DCOM), in favor of utilizing their own browser “plug-in” format and a flavor of Java Beans which supports a competing standard for distributed object brokering, CORBA. Likewise Microsoft is not supporting directly some of the Netscape features in this area. This generic capability (i.e., the automatic downloading and installing of software components), is essential to the NIA, so at the present time it is necessary to chose one over the other. For the Naval Aviation Community, it is thus recommended that the Microsoft Internet Explorer Web Browser be used for Naval Aviation IETMs, with its support of DCOM and, in particular, ActiveX controls. It is also recommended that a limited ongoing assessment and development

effort be sustained specifically to determine how and when Netscape can be brought up to generic DCOM capability through third-party plug-ins or a change in Netscape policy.

There is also a marked degree of difference in how the two products handle Dynamic HTML, an emerging technology for putting intelligence into actual Web pages. However, there are options for this functionality and the Naval Aviation Study team has not yet established this requirement as part of the minimal baseline. More time is needed for this technology to mature and to allow an assessment of the marketplace's support for a common approach.

The de facto level of functionality of these two primary browser products sets the minimal capability for most of the requirements; the basic capability of a browser product is typically very useful and it costs nothing to add it to a list of requirements. However, the incorporation of additional requirements would typically be very costly and would result in non-standard installations and thus would need justification. For most functions, therefore, the only real choice is to select one of the two commercially available products (Netscape Navigator and Microsoft Internet Explorer) or to require that IETM products operate with both.

The most complete solution to support of all four Architectural Types is the native support for the Microsoft DCOM family of distributed object. At this time Netscape, will not support DCOM. While it is arguable that Netscape provides a greater level of support for a widely divergent client base (i.e., many different computers and operating systems), the Navy policy to standardize on IT-21 and the Microsoft NT networking tools clearly favors the Microsoft DCOM standards and, accordingly, Internet Explorer. In an implementation of the NIA in the Naval Aviation Community, the only software that needs to be overtly loaded on the display devices for IETMs, other than the operating system and the personal servers for stand-alone usage, is Internet Explorer.

## **8. SERVER AND DATABASE INTERFACE SPECIFICATION**

Minimum server capabilities are highly dependent on the type of Architecture of the system being utilized. For Types 1 and 2, virtually any commercial HTTP server can be utilized. For Type 3 and Type 4 applications, the situation is not so straightforward because of the immature and emerging technology situation of the server marketplace today. There is economic pressure on software vendors to develop a competitive advantage (i.e., proprietary features) in their server products, since it is widely recognized that profits will be made only in the server market place. As long as Microsoft and Netscape continue to make their powerful browser product available free of cost, there is no money to be made in the browser market. Vendors will seek to make their profit in the Server market, and a direct result will be a rash of proprietary server products, a situation that does not lend itself to standardization of DoD servers.

Specific considerations for the server capability in a NIA compliant intranet are as follows.

## 8.1 Type 1 and Type 2 Server Support

Virtually any robust commercially available server product running on any Operating System will support Type 1 and Type 2 applications since all intelligent processing is performed in the browser and not on the server. All the server needs to do is serve out HTTP referenced files and Web Pages. For the Naval Aviation Community with its goal of achieving 100% IT-21-compatible servers (i.e., Windows NT Server), it is considered that the IIS version 4 server included with Windows NT Server is clearly the preferred choice.

## 8.2 Type 3 Server Support

The Type 3 Server Support requirement is a function of the Type 3 Architectural Type itself. There are several varieties of Type 3 applications that require extensions beyond what might be called a "vanilla" server. One approach is to use a proprietary server which, when it is installed, automatically provides a specific custom application software. Two examples of this type relevant to IETMs are the DynaWeb product, a Web-enabled version of the popular Dynatext electronic text viewer, and the TechSight Web product of General Dynamics. These products offer powerful functionality because of the server software, but have the distinctive problem of creating a life-cycle software maintenance requirement for the life of the product, since the software must be modified to upgrade the functionality of the IETM; that is, they do not have the commodity character of an out-of-the-box product such as the Microsoft IIS Server. Another variety of a Type 3 server extension is closer to this commodity situation and involves a standard set of server extensions such as Microsoft's Front Page and the Active Server Page (ASP) extensions to IIS, which are installed only once. Functionality is added to the server by means of server components that are included in the IETM Web-Page which are automatically installed on the server in a manner similar to the automatic component installation for the browser, as in the case of ActiveX controls. Microsoft does offer such components and the tool kits to develop them in its Visual Studio product (especially Visual InterDev) and a meaningful level of support in its low cost Front Page 98 product. They are, however, proprietary to Microsoft.

For the Naval Aviation Community and its commitment to IT-21 and the Windows NT technology, it is recommended that the basic server requirement be expanded to include the requirement to install the Microsoft Front Page 98 and the ASP extensions to the IIS 4.0 server included in the latest releases of Windows NT Server. These are no-cost options for the NT Server and no-cost upgrades are available for existing NT 4.0 installations.. They involve simply a commodity installation, which can be done once at server inception. The maintenance of the functionality in the IETM is entirely included (i.e., encapsulated in the IETM View Package as what Microsoft calls "Design Time Controls"). Any custom extensions beyond this would require the justification that it is needed for the entire Naval Aviation Community and should managed as such. In particular, such a situation may be justified for wholesale inclusion of a legacy capability into the NIA implementation, such as establishment of a capability to utilize the ATIS inventory of TM images.

### **8.3 Type 4 Server Support (Database Interface)**

Type 4 Architectural applications are those in which the content data is primarily resident in a database and the object encapsulation serves as organizing shells or templates. In fact, in an IETM for which the format has stabilized, there may be no need to modify the encapsulated objects when content changes are made. Only the database instance needs to be modified. Such a construct was envisioned when the "Class 4 IETM" was prototyped almost ten years ago. Virtually all database vendors are marketing a Web-enabled variant of their Database Management Systems (DBMS). This is an emerging area in which new products are being developed every month, many of which are applicable to IETMs. Thus, this is not the time to restrict or standardize the Type 4 solutions. More study and time are required. The specific recommendation for the Naval Aviation Community is to install the Microsoft ASP extensions which contain a set of preprogrammed interfaces to many popular DBMSs, employing the widely accepted ODBC interface standard. This is the same set of extensions recommended for Type 3 IETMs with the addition of preprogrammed specific DBMS interface packages. This approach would allow Type 4 developers to use DBMSs such as MS SQL-Server, Oracle, or even MS Access without requiring modifications to the AME servers. However, the next-generation object-oriented databases such as Versant will require a small but customized interface package and will need further study.

The approaches outlined in this report address general technology and object standards; they do not provide an assessment of individual commercial products. However, a vendor trend that is only recently emerging is to introduce attractive features in proprietary intranet-oriented server products especially in the use of DBMS products. The original concept, upon which the NIA effort was based in October 1996, was to allow any number of proprietary authoring products and associated methods to be encapsulated into the IETM objects, but to utilize only generic servers in the field. The Project is still holding to the bias of this approach but, during future study phases, it must investigate selected server-extension products that solve problems not easily solved by pure encapsulated objects. In particular, this situation occurs when a Type 4 IETM application requires the services of a separate DBMS as well as the presentation method that is encapsulated in the IETM object. In this case, it may not be feasible to force that DBMS into the encapsulated object. Other commercial server extensions will be examined in future studies, but, in general, these risk introduction of a high infrastructure cost and a situation involving proliferation of non-standard servers that is not in the interest of interoperability.

## **9. ELECTRONIC ADDRESSING SPECIFICATION**

Implementation of electronic addressing requires two things: (1) a mechanism and format for encoding electronic addresses into an IETM VP, and (2) a defined name space and address model. For use of the NIA, it is recommended that the Internet URL be the format for the address reference and the Industry standard practice of employing URL references for automated electronic linking be adopted. This practice makes URL references embedded in an electronic document visible as hot spots which cause the

default browser to open up a new page with the content of the referenced URL displayed in that page. Thus actual URL coding can be hidden under a more human-readable text string or graphic, as in the case of most Web-page authoring programs; or the URL itself can be made the hot spot as is the case with MS Office Products, such as MS Word 97. Additionally, the Electronic Addressing practice recommended will employ persistent URLs or P/URLs that, once established, remain the same no matter where, or on which intranet or server, the object resides. The intranet can re-map the notional server references to the actual server site on which the files exist using standard features such Domain Name Servers (DNS) or other server names to actual IP address mechanisms such as the Windows HOST file. In this light, the server name required below does not have to actually exist on the Internet as it will always be remapped onto an actual intranet server.

Guidance for establishment of P/URLs is documented in Figure 5.

**Figure 5 - Guidance for Establishing P/URLs in the NIA**

P/URLs will be authored and maintained as follows:

HTTP shall be the Web-page protocol to be utilized in this architecture (i.e., the URL starts with "HTTP://")

Server Name is to be in the form of "natsf.navy.mil" and is listed as though it were an actual server on INTERNET. It is recommended that management activities actually install such a server and maintain all of their cognizant URL references on that site in the form of acknowledgment as valid reference even if the actual content is not included for security reasons until such time that a secure DoD network is available.

File/Directory notation is to be unique across all DoD IETMs and is administratively assigned as though it were the IETM number in the form of a Unix file system reference with forward slashes such as "/navy/f18/ef/engine/ge/"

Additional Directory breakdown of files with in IETM] is merely a further extension of the Assigned File/Directory name and content for a section within an IETM and may be null for the top level reference.

Sample: "/navy/f18/ef/engine/ge/diagnosis/test3".

Optional user-defined moniker may be utilized These are most commonly used for carrying detailed data-base access parameters in the URL.

## **10. ASSURING AND TESTING COMPLIANCE**

The requirements of this Architecture are specific, and it is possible to prepare a checklist and list of criteria to determine whether the requirements have been met in a given case. However, the nature of the Architecture is to produce an IETM that actually operates in an interoperable manner in an actual installation. Accordingly, it is recommended that for Naval Aviation the primary compliance testing be based on establishing whether IETMs generated using the Architecture work in the recommended Microsoft NT/IIS/IE Version 4 Intranet environment. Such testing is easily replicable on PC's in the IETM creator's facility or at the Infrastructure receiving site.

A more difficult, but equally important, factor requiring compliance testing involves the electronic-link references. There are two aspects to this question:

- (1) With regard to the validity of the linkage, it must be established that the specific URL matches exist once the basic referenced document has been installed in the Intranet and, when a custom component has been automatically installed in the browser device, that the linkage methodology is working properly.
- (2) There is also a need to create a usage guide for the URL references which has not been fully developed at this time.

Certain easily avoidable practices that will operate in the Internet will not work well in a remappable NIA-compliant intranet that uses P/URLs. This included guidance will include requirements such as avoidance of fixed IP addresses and the use of relative and not absolute internal URL references in an IETM. The associated acceptance and compliance testing procedures will also have to be developed. Implementation of such a capability will, of course, be a major future task for the Naval Aviation Community and a necessary part of developing a capability to accepting IETMs from weapon system contractors.

## **11. MIGRATION AND INTEGRATION OF ELECTRONIC LEGACY SYSTEMS**

This Architecture has intentionally been designed to accommodate legacy systems with no modification of the legacy data format. However, establishment of this capability will require that the legacy presentation software be converted to be a Web-compatible software presentation component for those legacy systems that will not be replaced by alternative data format. For many legacy data formats the needed Web-capable components have already been developed. Examples are PDF, MS/WORD files, and most common graphic formats. However, some custom DoD IETM-presentation systems have not been converted. In these situations the current application would have to be converted into a form compatible with the Web browser such as an ActiveX control or a Java-beans application. More complex applications such as those utilizing a DBMS need to be converted to a Web-compatible system for the database access. The conversion effort is typically more difficult for an application that is programmed in an older 8- or 16-bit application language. Newer applications using 32-bit development tools,

especially those developed for Windows platforms, will experience much less software conversion effort since ActiveX is based on the earlier OLE standard used in most Microsoft Windows-targeted software applications. In other applications for which there is a large but not growing inventory of legacy material, it may be more appropriate to perform a one-time conversion of the data to a format more amenable to standard Web presentation. Such issues must be evaluated on a case-by-case basis.

## **12. IMPORTANT UNRESOLVED ISSUES NEEDING FURTHER STUDY**

A variety of technical issues and implementation issues within the original scope of this study remain. At this time the NAVAIR Study Team is not prepared to make formal recommendation without further study and evaluation. In many cases, more time is needed for the completion of Industry standards and the establishment of de facto standards in commercial practice, and for the emergence of mature vendor products. However, these details will be needed for a complete architecture and are listed here for information. There are also many related management and even technical issues outside the scope of this effort, which will need attention in the future. This involves areas such as the configuration management of the IETM View Packages in the many distributed data repositories (i.e., intranets) and the development of an electronic library model with various index and other metadata files and databases to facilitate access to the IETMs as the installed inventory gets very large.

### **12.1 Maintaining a Common Look and Feel among Differing IETMs**

While the use of the common browser does standardize many of the user-interaction features, it is very possible to include a custom component that contains its own set of unique user interaction features layered under the higher-level browser toolbars. These features often conform to a proprietary look-and-feel standard. A well-known example is the PDF Acrobat Viewer component, which includes all the Acrobat user features within the browser viewing window for features such as zoom, next page, scroll, etc.

A requirement still exists for a procurement guidance mechanism for minimizing the differences in Look-and-Feel among various disparate IETM presentation components that operate in the NIA environment. From both the Training and the Job Performance perspective, the effectiveness of each product is enhanced when they are displayed in accordance with a standard style, even if the actual underlying IETM presentation components vary and are proprietary in nature. The specific recommendation of this study is to revise the MIL-PRF-87268 specification to apply to the NIA framework and to make that revised specification available to IETM procurement officials as an acquisition tool. IETM TMCRs and other procurement instruments could then require that delivered IETM View Packages conform to both the NIA performance specifications and the revised MIL-PRF-87268 user-interface requirement for a common DoD IETM Look-and-Feel interface. The NIA specifications conform to the DoD Acquisition Reform initiatives. This effort should be coordinated DoD-wide as was the original MIL-87268

military specification. The requirement extends to a much larger community than just Naval Aviation.

## **12.2 Updating View Packages through the Navy Infrastructure**

Technology to manage and process IETM View Packages through an Infrastructure has been demonstrated in the IETM laboratory; however, more detailed Infrastructure design is limited by two factors. First, the Navy is in very early phases of designing such an Infrastructure and operational concepts have not yet been finalized. Second, the area of push technology and standards such as Microsoft's Channel Definition Format are not mature but are nevertheless universally recognized as the most effective basis for these functions. Both of these areas need further analysis as well as more time to mature.

## **12.3 Updating Type 4 IETM Implementations**

While the Type 4 Architectural Application is the most likely mature architecture for Class 4 IETMs (those based on PRF-87269-conformant databases), the technology and products to support it are immature and still emerging in the marketplace. The Study Team has recognized this Architecture Type as the best for future large-scale IETM applications and strongly recommends a continuing study effort for this area. A particular area needing continuing assessment is the updating of the database content. Most likely, the preferred way of updating these databases is to use the tools applicable to the DBMSs, most of which have proprietary data-replication facilities for this very purpose. While these are network-enabled, the data-replication facilities typically utilize network protocols and procedures different from that peculiar to the World Wide Web and, as such, not compliant with the NIA as described in this report. However, there is evidence of a strong Industry trend to blend these two technologies (Database and Web technology) and a high likelihood that industry practices will arise in the near future which should be applicable to the future NIA recommendations.

## **12.4 Automatic Component Installation**

A key aspect of the NIA design is that all software components be accessed as data from the server and automatically installed on the display device without user intervention. Technology to perform this function clearly exists and has been demonstrated using commercial products in the NSWCCD IETM Laboratory. However, the Web-based methodologies which easily achieve this feature (e.g., encapsulated in HTML using the OBJECT tag) and the preferred encapsulation (which may or may not be HTML) for the IETM View Packages may not always correspond. It is possible to employ administrative corrections to solve this apparent problem (e.g., in this case, require user to access an autoloading HTML object before using the native data object); however, they do require the user to exercise some discipline regarding the order in which some IETMs are viewed and are difficult to enforce in the user-device software. In final implementation it may also prove to be counterproductive to insist on a pure reading of the Architecture philosophy requiring only automatically downloadable components in

all circumstances and some workarounds may be needed to contain the cost of a specific implementation.

## **12.5 Implication of Non-Microsoft implementations**

As noted, a number of specific recommendations concerning the design of the NIA for the Naval Aviation Community have been strongly influenced and simplified by the Navy's decision to require IT-21 Compliant Systems. However, if this architecture is to be applicable to non-Microsoft environments (e.g., UNIX), some of the specific recommendations will not apply. Further study is clearly needed in cases where this situation occurs, as it will in the case of developing a DoD generalization of the architecture.

## **13. BUILDING INTEGRATED PRODUCT SUPPORT DATABASE (IPSDB) USING THE NAVY IETM ARCHITECTURE**

This report closes with the following recommendation for increasing the applicability of the NIA model to applications other than IETMs. The above-described Architecture can apply to any of the components of an Integrated Product Support Database (IPSDB), including training products used to support a weapon system in the field. In developing integrated support for a product, which includes IETMs as well as training modules, it should be the DoD position to discourage the development of proprietary monolithic IPSDBs for individual weapon systems. Instead, it is recommended that a strategy be developed for using the proposed unified IETM architecture to provide IPSDB functionality incorporating field technical training, diagnostics, and logistic support products. The family of general-purpose commercial products being developed for private intranet Web servers utilizing Internet World Wide Web technology can provide all the functionality needed and should be adopted instead of applying traditional customized application software approaches usually employed to develop custom DoD product-support systems.

## 14. ABBREVIATIONS

|         |  |
|---------|--|
| AME     | Automated Maintenance Environment                      |
| ASP     | Active Server Page                                     |
| CGI     | Common Gateway Interface <sup>1</sup>                  |
| CORBA   | Common Object Request Broker Architecture <sup>2</sup> |
| COTS    | Commercial-Off-The-Shelf                               |
| DNS     | Domain Name Service or Domain Name System <sup>3</sup> |
| DCOM    | Distributed Component Object Model <sup>4</sup>        |
| GCSS    | Global Combat Support System                           |
| GIF     | Graphics Interchange Format <sup>5</sup>               |
| HTML    | HyperText Markup Language                              |
| IETMTWG | IETM Technology Working Group                          |

---

<sup>1</sup> For performance considerations on using the CGI, see Internet URL at: <http://www.pcwebopaedia.com/CGI.htm>

<sup>2</sup> Short for Common Object Request Broker Architecture, an architecture that enables pieces of programs, called objects, to communicate with one another regardless of what programming language they were written in or what operating system they're running on. CORBA was developed by an industry consortium known as the ObjectManagement Group (OMG). See Internet URL: <http://www.pcwebopedia.com/CORBA.htm>

<sup>3</sup> DNS is an Internet service that translates domain names into IP addresses. See Internet URL: <http://www.pcwebopedia.com/DNS.htm> for a list of Internet Resource dealing with DNS.

<sup>4</sup> Short for Distributed Component Object Model, an extension of the Component Object Model (COM) to support objects distributed across a network. DCOM was developed by Microsoft and has been submitted to the IETF as a draft standard. Since 1996, it has been part of Windows NT, and is also available for Windows 95. See Internet URL: <http://www.pcwebopedia.com/DCOM.htm>

<sup>5</sup> Technically, a GIF uses the 2D raster data type, is encoded in binary, and uses LZW compression. There are two versions of the format, 87a and 89a. Version 89a (July, 1989) allows for the possibility of an animated GIF, which is a short sequence of images within a single GIF file. A GIF89a can also be specified for interlaced presentation.. A patent-free replacement for the GIF, the PNG format, has been developed by an Internet committee and major browsers will soon be supporting it.

|        |   |
|--------|---|
| IIS    | Internet Information Server <sup>6</sup>                  |
| JCG-CE | Joint Commanders Group for Communications and Electronics |
| JLC    | Joint Logistics Commanders                                |
| JPEG   | Joint Photographic Experts Group <sup>7</sup>             |
| MIME   | Multipurpose Internet Mail Extensions <sup>8</sup>        |
| NIA    | Navy IETM Architecture                                    |
| NLISP  | Navy Logistics Information Strategic Plan                 |
| ODBC   | Open Data Base Connection                                 |
| PEDD   | Portable Electronic Display Device                        |
| PDF    | Portable Data Format <sup>9</sup>                         |
| PNG    | Portable Network Graphics <sup>10</sup>                   |
| PURL   | Persistent URL <sup>11</sup>                              |

---

<sup>6</sup> IIS is a Web Server from Microsoft. It is a component of Microsoft's Windows NT 4.0 Server Operating System.

<sup>7</sup> A JPEG (pronounced JAY-peg) is a graphic image created by choosing from a range of compression qualities (actually, from one of a suite of compression algorithms). Along with the Graphic Interchange Format (GIF) file, the JPEG is a file type supported by the World Wide Web protocol, usually with the file suffix of ".jpg". You can create a progressive JPEG that is similar to an interlaced GIF.

<sup>8</sup> MIME, a specification for formatting non-ASCII messages so that they can be sent over the Internet. Many e-mail clients now support MIME, which enables them to send and receive graphics, audio, and video files via the Internet mail system. here are many predefined MIME types, such as GIF graphics files and PostScript files. It is also possible to define your own MIME types. n addition to e-mail applications, Web browsers also support various MIME types. This enables the browser to display or output files that are not in HTML format. See Internet URL: <http://www.pcwebopedia.com/MIME.htm>  
The MIME related Requests for Comments may be found at Internet URL: <http://www.oac.uci.edu/indiv/ehood/MIME/MIME.html>

<sup>9</sup> Adobe's Neutral Data Format for documents. The Reader is free from Adobe Systems at: <http://www.adobe.com>.

<sup>10</sup> PNG (pronounced "PING") is a file format for compressed graphic images that, in time, is expected to replace the GIF format that is widely used on today's Internet. The GIF format is patented by CompuServe and its usage in image-handling software involves licensing or other legal considerations.

<sup>11</sup> Short for Persistent URL, a type of URL that acts as an intermediary for a real URL of Web resource. When you enter a PURL in a browser, the browser sends the page request to a PURL server which then returns the real URL of the page. PURLs are persistent because once a PURL is established, it never needs to change. The real address of the web page may change but the PURL remains the same. See

|        |  |
|--------|--|
| TCP/IP | Transmission Control Protocol/Internet Protocols <sup>12</sup> |
| URL    | Uniform Resource Location. <sup>13</sup>                       |
| W3C    | World Wide Web Consortium <sup>14</sup>                        |

## 15. GLOSSARY

**Active Server Pages** is an open, compile-free application environment in which you can combine HTML, scripts, and reusable ActiveX server components to create dynamic and powerful Web-based business solutions. Active Server Pages enables server side scripting for IIS with native support for both VBScript and Jscript.<sup>15</sup>

**Design\_Time Web Controls** Design-time Web controls are standard ActiveX&trade; controls that have a special interface, called IActiveDesigner, that lets them generate text that is saved into a file by the editor and processed at runtime. The structure and content of the text that is generated is entirely up to the control—any text that can be inserted into the file by a standard text editor can be generated by a design-time control. Design-time controls are based on COM, so they're easy to build, share, and host. They also differ from standard ActiveX controls in that they

---

Internet URL: <http://www.pcwebopedia.com/PURL.htm>

<sup>12</sup> These are the basic communication protocols for the Internet. See: Postel, J., "Internet Protocol - DARPA Internet Program Protocol Specification", RFC 791, DARPA, September 1981; Postel, J., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, DARPA, September 1981.

<sup>13</sup> A URL (Uniform Resource Locator) (pronounced "you-are-EL" or, in some quarters, "earl") is the address of a file or other resource accessible on the Internet. The type of file or resource depends on the Internet application protocol. Using the World Wide Web's protocol, the Hypertext Transfer Protocol (HTTP), the file can be an HTML page (like the one you're reading), an image file, a program such as a CGI application or Java applet, or any other file supported by HTTP. The URL or resource address includes the name of the protocol required to access the file or resource, a domain name and, if it's a file, a hierarchical description of a file location on the server. Source: Internet URL: <http://whatis.com/url.htm> and RFC 1738 at Interent URL: : <http://andrew2.andrew.cmu.edu/rfc/rfc1738.html>

<sup>14</sup> W3C Home Page. Internet URL: <http://www.w3.org/>

<sup>15</sup> Microsoft Site Builder Network Feature Stories. Internet URL: <http://www.microsoft.com/sitebuilder/archive/features/aspovert.htm>

contain no binary runtime component—they're never "alive" when a page is being viewed. Instead, a user sees only their HTML output.<sup>16</sup>

**Frames** A feature supported by most modern Web browsers that enables the Web author to divide the browser display area into two or more sections (frames). The contents of each frame are taken from a different Web page. Frames provide great flexibility in designing Web pages, but many designers avoid them because they are supported unevenly by current browsers.<sup>17</sup>

**JavaBean** JavaBeans is the platform-neutral, component architecture for Java. JavaBeans allows developers to create reusable software components that can then be assembled together using visual application builder tools, such as Sybase's PowerJTM, Borland's JBuilderTM, IBM's Visual AgeTM for Java, SunSoft's Java WorkshopTM and Symantec's Visual Cafe, and many others.<sup>18</sup>

---

<sup>16</sup> March, 1997, Microsoft Interactive Developer Column: Preview by Jay Massena. Internet URL: <http://www.microsoft.com/mind/0397/preview0397.htm>

<sup>17</sup> Internet URL: <http://www.pcwebopedia.com/frames.htm>

<sup>18</sup> JavaBeans : The Only Component Architecture for Java™. Internet URL: <http://www.javasoft.com/beans/>

This Page Intentionally Left Blank