AD-A241 062

(2)

Technical Document 2116
July 1991

White Paper on the Database Management System Interface Standard for Navy Next Generation Computing Resources (NGCR)

D. L. Small
M. C. Butterbrodt
R. M. Bergman

NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152-5000

J. D. FONTANA, CAPT, USN
Commander

R. T. SHEARER, Acting
Technical Director

ADMINISTRATIVE INFORMATION

The study covered in this document was performed from November 1989 through May 1991. It was sponsored by the Space and Naval Warfare Systems Command. Washington, DC 20363-5100. The funding was under program element OMN, project no. 41-CC30 01, work unit DN587574, and NOSC job order number CC30411F03. The work was performed by Code 413 of the Naval Ocean Systems Center (NOSC), San Diego, California.

Released by
M. C. Butterbrodt, Head
Distributed Systems
Branch

Under authority of
A. G. Justice, Head
Information Processing and
Displaying Division

RT

CONTENTS

FIGURES

1.0 BACKGROUND

The charter of the Navy's Next Generation Computing Resources (NGCR) program is to provide the Navy Mission-Critical Computing Resource (MCCR) applications with a coordinated set of interface standards for both physical and logical computer resources. These standardized interfaces will improve industry's ability to provide computing resources that meet Navy needs. The interface standards are to be widely accepted, non-proprietary, and, if possible, widely used within industry.

The Database Management System Interface (DBMSIF) standard, the subject of this document, is one of the set of standards that is essential to the timely and cost-effective acquisition of the majority of the next generation of Navy mission-critical computing systems. The DBMSIF will support the Navy in efficiently acquiring systems that address a wide range of performance, compatible computing service, and functionality.

2.0 DBMSIF WHITE PAPER OBJECTIVE

This technical document is a white paper that presents the advanced analysis, investigation, and planning regarding the issues facing the NGCR Database Management System Interface Working Group (DBWG) and provides a starting point for discussions when the group is initiated in FY 92. This white paper is a "snapshot" of the Navy's use of database management system (DBMS) technology for command, control, and combat systems as of this writing. The long-term goal is to contribute to the development of a standard DBMS interface to promote interoperability among Navy systems. This document is a required deliverable under current Naval Ocean Systems Center (NOSC) tasking for the NGCR Program.*

* This work was supported by SPAWAR 3243, the Next Generation Computing Resources (NGCR) program, and will be available in electronic mail format as well as in a NOSC technical document.

3.0 NGCR PROJECT SCOPE

The NGCR interface standards set, while being developed incrementally is to be sufficiently in place so that the Navy can begin acquiring systems using these standards by 1998. The work being performed to develop these standards is being offered to industry and academia to present the Navy's directions for systems development.

The task of DBMSIF standards development will begin in FY 91 and continue through FY 98. The initial DBMSIF standards will be available for use in acquisitions starting in FY 95.

The initial range of applications includes computing from the single dedicated processor to networked, heterogeneous, modularized, backplane, bus-architecture computing systems. Networking is to be accomplished using NGCR Local Area Network (LAN) standards and, as appropriate, other MIL-STD networking interfaces.

4.0 SCOPE OF DBMSIF ENVIRONMENT

The scope of the environment—and therefore of the resulting standards that the DBMSIF standard should support—must be carefully considered. A representative list of questions concerning this follows:

1. How should the interface to a realtime operating system for resource management (including control of memory location, network scheduling, and configuration of internal and off-line memory) be considered?

2. Should the hardware on which the DBMS is used make a difference as to whether it is one machine or a network of machines with transparency as to location of data? Should it make a difference as to the type of machine (parallel processor, multiprocessor, ...)?

3. Should distribution of data make a difference and how does it make a difference (global/local data considerations)?

The answers to questions 1 through 3 could imply that there is only one scheduling interface for how a database system interacts within the NGCR system. There may be options, such as time requirements, resources required, location of resources (local or over a network), and so on.

4. How is security supported by the DBMSIF?

5. How are fault tolerance and recovery supported by the DBMSIF?

6. How is the total life cycle of data supported by the DBMSIF?

7. Where and how do the impacts of questions 4, 5, and 6 differ? At least one of the similarities appears to be the need for timely maintenance of consistent versions of available data.

8. Should the kind of DBMS used or the size of the database make a difference—flat files, heterogeneous, hierarchical, relational, object oriented?

9. What impact do programming languages have on the DBMSIF?

10. Is one method of accessing the data reasonable, i.e., the Structured Query Language (SQL); or should there be other methods, such as use of an object-oriented (OO) language?

11. Should standardization occur at the SQL or logical interface only?

12. Should naming conventions (i.e., naming as used by directories or naming for database entries) be part of the standard?

13. What methods should be used to access remote distributed heterogeneous data? Two examples are ANSI/ISO RDA (American National Standards Institute/International Standards Organization Remote Data Access) and distributed object-oriented data access.

14. How will interfacing to knowledge-based systems impact such a standard?

The answers to these questions and others will help determine the scope of the DBMSIF environment within the NGCR system. Topics to address include at least how rigid the access to the operating system should be for coordinated resource management (don't always rely on data being provided by local memory), what protocols should be used or can any be streamlined for interfacing to distributed heterogeneous database systems to meet scheduling requirements, and what type of DBMS access will provide secure and reliable data. If the interface is designed with performance and transparency in mind, then, a data request is a data request, regardless of whether the request is internal, local, global .... Consideration will be given to adaptability and scaleability of the standard (i.e., different types of data and database concepts that must be accommodated, i.e., the extent to which the standard can be scaled to the job required).

5.0 NAVY REQUIREMENTS FOR DBMS INTERFACE

Navy systems have a requirement for managing massive command, control, communications, and intelligence (C3I) systems encompassing land, surface, subsurface, air, and space data elements. These systems ultimately control thousands of complex sensor, combat direction, and weapon systems aboard hundreds of tactical units. Driving such systems are significant requirements for managing such objects, discriminating the real threats among them, and tracking them with realtime updates using an intelligent analysis of which objects are benign (friendly, neutral, or decoys) and which are threats. The systems are necessarily distributed and require substantial data that must be consistent through time, often requiring a hard, realtime deadline to be met, based upon data availability and accessibility. To succeed, a thorough, consistent, and logical data model must be used for all dispersed components of the Navy's C3I and combat systems. The model must be based on multiple large disparate databases, containing common information requiring timely, consistent, and uniform access (see figure 1).

The Navy requirements for DBMS and the standards for its use have been, at least, in the strategic areas, very informal to almost ad hoc; that is to say, very project-requirements oriented. Two universal reasons for this are performance (very slow and cumbersome), and memory (high memory budgets are required for both internal memory of computers and external storage). A third reason is the lack of formal operating systems or operating system interfaces for a DBMS to "hook to" in such systems. Tactical weapon systems require stringent performance within the realtime-to-critical-realtime performance envelopes. In such cases, there are hard deadlines to meet with only a finite amount of processing time available. Two examples of this are as follows:

1. Threats being faced by a carrier battle group with a saturation raid of tactical aircraft that could be compounded by standoff jammers degrading sensor performance and/or cruise missiles being fired from different types of platforms.

2. Response to identifying low observable aircraft and enabling interception before they reach their targets or go beyond range.

In all such cases, optimal computer systems performance and interoperability are rerquired for engagement responsiveness for detection, classification, and scheduling. To meet this type of threat-performance requirement, most of the war-fighting "computer code" running in Navy systems today uses a significant level of hand-tailored assembly-level code, rather than using a set of standardized interface tools for managing data and scheduling resources. Such code also does not provide for an interface between responsive target management and identification of the likely target point of origin where such data probably reside in a large "unresponsive" database system.

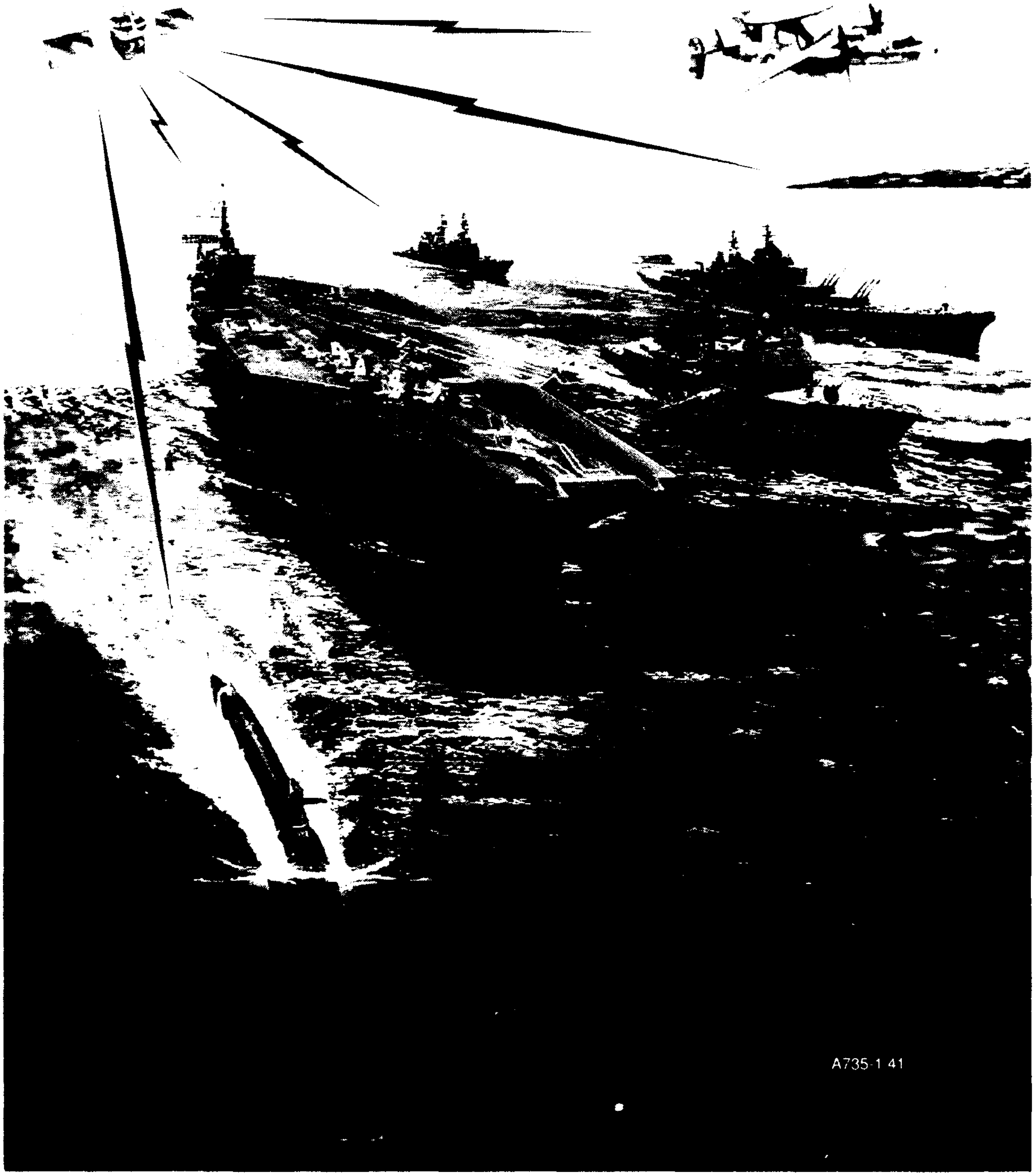

Figure 1. Navy (C3I) systems.

Personnel, logistics, and some of the strategic systems use a wide variety of commercial hardware and software to perform their not-so-realtime missions. These systems do use commercial operating systems and DBMSs. Their entire package of mission code does not have to be installed on platform mission computers. These types of systems are primarily batch (sequential operation) systems oriented, very rigid in their processing requirements and not specifically subject to the rigors of performing asynchronous external events to which NGCR systems must respond.

The Navy's DBMS interface requirements will be formulated by the DBWG converging in early FY 92. This group, as did the working groups preceding this one, will develop the requirements with joint government, industry, and academic participants. In advance of this effort, a number of well-understood high-level aspects and areas of requirements will be discussed in the following chapters.

6.0 BACKGROUND ON NAVY USE OF DATABASE MANAGEMENT SYSTEMS

In the strictest interpretation of NGCR development of standards, the DBMS standard shall be an Interface standard. The objectives for standardizing a DBMSIF are to promote application system portability, interoperability, and software maintenance and reuse, as well as a more common and meaningful representation of data throughout a system.

Most of the currently deployed tactical weapons and sensor systems (especially the systems utilizing the Navy standard computers, and CMS-2 and other pre-Ada languages) do "data management," but not with formal DBMS structures. These software structures are, for the most part, handwritten and tailored around very primitive and fundamental "executives" and "kernels" used as operating systems. These are used for two main reasons: (1) DBMS structures did not exist as part of the Navy/Marine Corps standard software products when most of these systems were designed and built (and still don't for the languages used for these systems) and (2) performance in critical time (hard deadline) situations is of primary concern. Only within the last year has Ada started to provide within its program library the functionality of the SQL language. The DBMS issue is still approached as application software that will meet the the mission requirements for which the system is being designed and built. An exception is the LHA amphibious ships' use of the Management Information System (MIS) as a general purpose data storage, processing, and retrieval system with application to Navy administrative, tactical, and strategic operations. The hierarchical database structure was used for the MIS on the AN/UYK-7 computer system because of a "nonmeasured" belief about its performance. The MIS system, for example, is used for setting up the planning for amphibious operations. After deployment, independent processing with the same computer system can be performed using the Tactical Data System (TDS) with all track flat files maintained in memory. Note here that MIS planning is carried out primarily before "time critical" operations take place.

6.1 DBMS UTILIZATION IN TACTICAL SYSTEMS

Target information has routinely been supported by linear flat files of tracks identified formally in message format sent between ships in tactical systems. With the advent of the ACDS, a linear file with a minimal number of attributes apparently is not adequate for today's track file management. An Object-Oriented (OO) data system is being used for threat management for ACDS to assist in manipulating some of these extra attributes.

In the Distributed C^2 Project at NOSC, an experiment was conducted using Naval Tactical Data System (NTDS) Link-11 data and the Oracle relational DBMS. The objective was to populate ORACLE Version 6 with Link 11 "nonrelational" data. The ORACLE system more than supported the data fill operation at the data rate provided. For a detailed discussion of the experiment and the accompanying lessons learned, refer to Butterbrodt (1988a), (1988b), and (1988c).

6.2 DBMS UTILIZATION IN STRATEGIC SYSTEMS

The strategic world has taken maximum advantage of the availability of commercial DBMS products. The strategic environment uses commercial hardware and software more widely because it enjoys relaxed physical, environmental, and critical-time performance constraints.

The Navy World Wide Military Command Control System (WWMCCS) Standard System (NWSS) first started by using the network database model to manage the various files for which CINCLANTFLT and CINCPACFLT have responsibility (e.g., FORSTAT, MOVEREP, equipment capabilities, and personnel information). The actual database was built using message-format design in an early version of NWSS. In the mid-70s, experiments were performed to move these files into a relational database format (primarily because of the Advanced Command Control Architectural Testbed [ACCAT] program). Due to the success of those experiments, the Navy is now beginning to convert present nonrelational Navy systems over to relational database systems.

A major example is the Naval Warfare Tactical Database (NWTDB) and a variety of smaller-scale systems that deal with data elements not covered by NWTDB. Another example is the Fleet Command Control Battle Management Program (FCCBMP) at CINCPACFLT currently using the ORACLE relational DBMS for managing its data. Now all WWMCCS and NWSS sites, including the NWSS sites, will be using the M204 database system built by the Computer Corporation of America (CCA). The M204 database system is an IBM hierarchical data model that is being converted over to a relational DBMS, based on the SQL standard.

Almost all instances of strategic systems, even though some are following the relational DBMS, are built on different processors and have different conventions for access and "nice to have" extensions that must be accommodated. Because of the conventions followed in network database systems, early conversion to relational systems used network links as column entries in relations—an easy technique to follow, but with resultant performance degradation. Only within the last few years has the relational system design been reconsidered. The technique of indexed columns—a non-SQL standard—is an extension used to improve performance by allowing access to only a few columns, instead of the large number considered for describing a relation entity. The cost is a larger memory budget.

The Navy, in its Naval Tactical Command System-Afloat (NTCS-A) program, is intending to use not only NWTDB and some other smaller static relational systems but also a static system called the Military Intelligence Integrated Database System (MIIDS). Within NTCS-A, these strategic static databases can be combined with the more tactical realtime ACDS. The MIIDS system is being put together by the Defense Intelligence Agency (DIA) as a series of flat files that are to be stored in the M204 database system.

The "relational design" consists of entities, such as personnel described by attributes, i.e., relational columns. The columns may or may not be filled and can be as many as 150 or more. Relationships between entities are defined, but those relationships are not contained in the data tapes maintained by DIA. The relational database standard is suggested, but performance will suffer if data are used in that form. Here, as is the case for ACDS, there is a significant issue of management of realtime data consistency while querying large databases (upwards of gigabytes in size in raw data form for MIIDS data). For ACDS identification data, WWMCCS/NWSS data, and NWTDB, there is the issue of how to design the relational system so that retrieval performance does not suffer. Techniques used in older systems will not necessarily allow for good performance for database access for relational systems with a much greater volume of data. (Such techniques use the network database model as available in WWMCCS—and linear files as used for processing NTDS/ACDS tracks.) These issues require detailed analysis to develop a versatile, adaptable interface standard.

6.3 DBMS UTILIZATION IN INTELLIGENCE SYSTEMS

The same type of issues, noted above for strategic systems, are being faced in the Navy's Intelligence Systems. In these cases, more data have to be handled, much of which is more free-form, textual, and static. The NTCS-A strategic system is the most similar in having to deal with such problems for static databases. These systems also face extreme data fusion requirements.

For all classes of systems, a number of smaller database systems are being used on Personal Computers (PCs), many of which use Ashton Tate's DBASEIII, IV, and V (also relational database systems). If there is not a standard DBMS interface to the NGCR system, the number and types of DBMS' will proliferate at an increasing rate—reference ACDS and the Naval Tactical Command Systems-Afloat (NTCS-A) as examples. This list continues with the systems that NTCS-A is likely to be deployed on and that is a large group—carriers, amphibious ships, cruisers, and perhaps smaller platforms. Each of these could be using different DBMS', but all require a standard DBMSIF.

7.0 INTERFACE STANDARD NEEDS

Based on the requirements of consistent, realtime, distributed, and heterogeneous Navy DBMS' providing application system portability, a number of interface needs must be established.

7.1 REAL/CRITICAL TIME

The interface standard must be implementable in ways that support the needs of real-time/critical time (hard deadline) systems. This implies that the interface must be as simple as possible and must accommodate and support a variety of scheduling approaches, e.g., changing scheduling priorities and what processes are to be scheduled for controlling resources so that a hard deadline can be met when using a processor and/or a network. The interface must be designed so that realtime system developers can configure the operating system and DBMS interoperability to best meet requirements. This includes managing data consistency and correctness with realtime updates for all sizes of databases. The Portable Operating System Interface for Computer Environments (POSIX) standard for Realtime Extensions (P1003.4) should be monitored. This standard will support the portability of applications with realtime requirements. Application Environment Profiles (AEPs) are already being supported in the P1003.13 AEP Working Group to identify critical-time requirements. For a further description of the POSIX family of standards (now selected as the NGCR operating system [OS]), refer to UniForum, 1989; and to the NGCR OSSWG Recommendation Report, May 1990.

7.2 HETEROGENEOUS

The DBMS interface standard must be implementable on a wide variety of hardware architectures, configurations, and capabilities, because more than one methodology and vendor's DBMS may be involved (note the success of the ORACLE relational DBMS on a variety of different computer platforms). This requirement pertains not only to the DBMS resident on a single processor, but also to a DBMS that spans multiple heterogeneous processors. The automatic exchange of information between heterogeneous and multimedia (text, graphics, images, and sound) is also an issue. Software requirements can include access to flat files, network, hierarchical, and object-oriented databases, as well as to relational databases, many of which are very large, yet require instant hard-deadline access to critical data elements. These systems can be based on the Navy's AN/UYK computer systems; the Enhanced Modular Signal Processor, AN/UYS-2 (EMSP); SUN Microprocessors; VAXs; PCs; embedded processors, e.g., 68030 boards; or parallel processing machines, such as the ENCORE processing system. This explanation should not preclude the possibility that some parts of the DBMSIF could actually be implemented in hardware, itself, to satisfy extreme performance requirements.

Note that all query languages among relational DBMSs are not alike. Most vendor SQL implementations vary in SQL support (built-in functions, relational operators), SQL syntax, SQL semantics (return codes, data type handling), transaction handling (commit processing, concurrency control, data isolation options), and data dictionary format. To date, there are no solutions to this problem. However, there is a clear industry trend toward both heterogeneity and distribution. Note, however, that NIST (National Institute of Standards and Technology) has recently approved the SQL Standard; refer to ANSI X3.135-1986 (ANSI, 1986).

Relational DBMSs can be CPU-intensive, if memory is appropriately scheduled, because multiple processes (searches, updates, additions, and deletions of records) all compete for processor time. Faced with multiple simultaneous requests, uniprocessors consume valuable resources for scheduling tasks. Parallel architectures will help distribute transaction loads simultaneously across multiple processors, permitting use of numerous high-performance processors to complete multiple tasks concurrently and, thus, yield faster response times.

7.3 DISTRIBUTED SYSTEMS

Distributed systems will continue to gain importance and application within the Navy into the 1990s and beyond. The data management issues in distributed systems will clearly be among the more difficult issues to resolve. The data types and volume of data, data currency, data consistency among parallel processes, along with the data fusion issues, stress the need for a DBMS interface standard for future success in Navy NGCR systems. The standard must support a vertical and horizontal hierarchy across systems, scaleable from the simplest to the most complex Navy systems, and still provide required performance. The primary objective of a distributed DBMS is to give interactive query users and application programs transparent access to remote data as well as to local data. An example of this would be if a track file (T) is located in the Naval Tactical Data System (NTDS) and the Advanced Combat Direction System (ACDS), and a ship's file (S) is located in the Naval Warfare Tactical Database (NWTDB). A distributed database (DDB) would allow a user located anywhere on the shipboard network to physically or logically enter a SQL statement to access data from the T and S files. The methods used to access remote data within the ship could conform with the emerging Open Systems Interconnection (OSI) Standard (Mollet, 1990), or they could conform to SAFENET's (Survivable Adaptable Fiber Optic Embedded Network) lightweight protocol suite for real-time applications.

7.4 LANGUAGE INDEPENDENCE

The DBMS interface must be defined generically; that is, it must be independent from any particular programming language. The services of the DBMS must be accessible by

Ada as well as by other programming languages in common use in Navy system applications, such as C, COBOL (Common Business Oriented Programming Language), FORTRAN (Formula Translation Programming Language), CMS-2 (Compiler Monitor System, Version 2), the Navy's standard programming language prior to Ada, artificial intelligence languages, natural language front ends, and languages used for signal processing.

7.5 OPERATING SYSTEM INDEPENDENCE

The DBMS interface standard should interoperate with the POSIX standard for the operating system interface (POSIX 1003/UniForum, 1990/NGCR OSSWG Report). P1003.1 defines the interface between portable application programs and the operating system, and supports application portability at the source-code level. This operating system standard interface will allow programs to be written for a target environment in which they can be ported to a variety of systems. A DBMS should be able to request resource management by interfacing with applications that use other operating systems.

7.6 NETWORK INDEPENDENCE

Ideally, there should be an interface with a wide variety of network architectures that is transparent to the DBMS interface. This network interface might conform to POSIX 1003.8-Network Services (UniForum, 1989). Such a use could permit transparent sharing of distributed files across systems. The DBMSIF should support media- and protocol-independent applications, and be consistent with existing and emerging standards, such as Open Systems Interconnection (OSI [Boland, 1989]). The interface should operate with local area networks (e.g., SAFENET, 1990) and wide-area networks. It should interface with a variety of different network architectures (e.g., OSI, SAFENET, Transmission Control Protocol/Internet Protocol [TCP/IP], Xpress Transfer Protocol [XTP] [Saunders & Weaver, 1990], Systems Network Architecture [SNA], and Digital Equipment Corporation Network [DECNET]).

7.7 DBMS INDEPENDENCE

The DBMSIF standard should provide DBMS interface independence whether or not the data model must use flat files; or hierarchical, network, relational, or object-oriented data.

7.8 SECURITY

Multilevel security and related issues (such as integrity) are critical to Navy systems. The DBMSIF must provide inherent security considerations within the interface design. POSIX 1003.6 (UniForum, 1989) is concerned with developing specifications for standard interfaces to security services and mechanisms for portable applications to include system call interfaces and system commands. The concern for multilevel security arises when a Navy computer system contains information with a variety of classifications (e.g., SCI, Top Secret, Secret, Unclassified) and has some users who are not cleared for the highest classification of data contained in the system. Since this feature cannot be added later, the DBMSIF must accommodate security considerations in all of its basic concepts. Discretionary and mandatory access controls must be specified at a minimum. Security features should be addressed that fulfill related requirements in the DoD Trusted Computer System Evaluation Criteria (DoD 5200.28-STD), commonly referred to as the "Orange Book"; and those in the Trusted Database Interpretation, when published by NCSC (National Computer Security Center) (NCSC, 1990).

7.9 FAULT TOLERANCE/RELIABILITY

Fault tolerance/reliability is an increasingly important area where DBMSs are expected to provide support. Fault tolerance and reconfiguration are of prime concern in Navy tactical systems for mission effectiveness. Redundancy and multiple points of access/connection can be major considerations for such systems. If parts of a system are damaged by either accident or combat, the system must maintain an operating status to support the particular mission. Data currency, consistency, and completeness are as important—or maybe more so—than the communications paths used for access. Optimal update strategies are needed when confronted with such circumstances.

The interface standard must support the capability to be creative in the above mentioned areas in distributed, heterogeneous systems when designing for fault-tolerant, dynamically reconfigurable modes of operation. One of the most important areas of resource management is the integrity of the data required to perform the mission. Data loss, quality, and accessibility are the essence of recovery from interrupted service or operation in degraded modes of a system. The interface standard and subsequent products developed against the standard must be sensitive to these issues.

7.10 DBMS LOGISTICS SUPPORT

A major issue is preservation of data, even though the database system being used may change or new functions (in the "data maintenance" mode) be made to work on the data. A good example of this is the effort the Navy puts into the maintenance of the Naval Warfare Tactical Database System. In providing such changes, data to be entered in the database system must not become contaminated and data must not become lost or stale (out-of-date). An on-line database has been added in which operators have access for change without contaminating the residence library (i.e, NWTDB).

8.0 CURRENT/FUTURE INDUSTRY STANDARDIZATION EFFORTS

8.1 CURRENT EFFORTS

The primary thrust of industry standardization efforts for DBMSIF is in the commercial world of data processing. The earliest of these efforts, which is still in use today, is the CODASYL database standard for the COBOL data-processing world. This has been a metric since the early 70s by which database (and language) products sold to the government for use in classical data processing are interfaced, measured, and accepted for procurement in the Automatic Data Processing (ADP) and nonrealtime areas of application.

To facilitate the use of databases, industry devised the Structured Query Language (SQL) as an interface to the relational DBMS—first introduced by IBM. Early use of it was encumbered by lack of performance, mostly due to lack of optimization and strict adherence to the table definition for relations and maintenance of Normal form, i.e., each attribute value in a row of a relation is atomic (nondecomposable so far as the system is concerned). This usage required large tables to be used that must be completely searched, unless a column, i.e., attribute, can be indexed and ordered to take advantage of a rapid search-retrieval mechanism.

Several groups are now developing standards to promote database interoperability: the American National Standards Institute (ANSI) Database Committee (X3H2), International Standards Organization/Open Systems Interconnection (ISO/OSI [Boland, 1989]), NIST, Open Systems Foundation (OSF), the SQL Access Group, and X/Open Company, Ltd.

During the last decade, the SQL has evolved to become the standard sublanguage for defining and manipulating data managed by relational DBMSs. The most widely known and accepted SQL standard is the one developed by the Database Committee (X3H2) of the ANSI. The ANSI SQL standard defines two language levels: SQL2—the complete standard, and SQL1—a subset of Level 2 (defined to be the intersection of existing implementations). For a complete description of the standard, refer to ANSI V8.135-1986. Other standards organizations have adopted ANSI SQL. To mention a few, the ISO accepted ANSI SQL in 1987; and in 1988, the U.S. Government added ANSI SQL to its Federal Information Processing Standard as FIPS-127. As we move forward into SQL3, many of the issues that exist today with SQL will be resolved.

Several major DoD projects and commands have selected the Oracle and SQL DBMS product as their DBMS of choice. These include the Naval Air Systems Command Software Engineering Environment (NASEE) Toolset, NAVAIR 546, and the Joint Integrated Avionics Working Group (JIAWG) Core Toolset. This product is also the prime choice of many other government agencies and government contractors because of its ease of use, documentation support, machine and environment independence, and abundance of software tools that can provide both logical and data structure interface.

8.2 FUTURE EFFORTS

Information Management

The relational DBMS and SQL query language are now in popular use by industry. But the complexity of building a SQL query has created a need for a new capability called the Fourth Generation Language for database systems (refer to ACDS' use of object-oriented databases). This capability is needed to describe a nonprocedural way to retrieve information without using the formal details of a SQL query by the user of a database system, e.g., "Find airfields located between latitude and longitude coordinates of 19 and 31 degrees North and 5 and 63 degrees East" as opposed to making that same query with SQL procedural commands that look like:

```
"select COORD_LAT, COORD_LONG, NAME_FACILITY
 from SITE_AFLD_INSTAL
 where (COORD_LAT Like '%N')
 And (COORD_LONG Like '%E')
 And (COORD_LAT >= '190000N'
 And (COORD_LAT <= '310000N')
 And (COORD_LONG >= '00500000E'
 And (COORD_LONG <= '0630000E')"
```

So far, no industry standards have evolved for this type of query language capability, yet a need for this capability clearly exists to preclude having to take a major class to use DBMS operationally.

Object-Oriented DBMS

Fourth generation language database capabilities combined with Object-Oriented (OO) Data Base Systems could address the complexities of SQL querying of data. The beginnings of such systems are starting to be provided in a rudimentary form by at least the SYBASE and PROGRESS DBMS', the Tigre Object Systems Company (Tigre, 1990),* and the Object Design Company with the product ObjectStore (ObjectStore, 1990). In the example above, airfields could be considered objects with descriptive attributes of latitude and longitude and the action of "find location."

"Management Information Systems (MISs) use Object-Oriented Databases (OODBs) with Fourth Generation Language DBMS', since they are the only systems that can really be used with OO languages. The ability to access this data from the "C" programming language would be preferable, but it is hard to standardize an OODB because of the difficulty of standardizing the format of the objects that are to be stored in it. However,

* Private communication: E-Mail message, dated May 1990, to Patricia Oberndorf, at NADC, from Jordan Bortz, President, Tigre Object Systems.

all OODBs should provide a reasonably complete set of access tools, and these access mechanisms may be standardized. An OODB requires the support of many OO design and implementation tools to easily construct applications and interface screens and to store data. Tools are needed as well for multiuser and networking support, transaction logging, and automatic garbage collection of DBs" (Tigre, 1990). In the flavor of this "white paper," resource management and access interfaces would be transparent to object manipulation.

Knowledge-Based Systems

Knowledge-based systems can be considered as an extension of OODB systems that have been used successfully in industry for manufacturing aids and sales distribution planning (Harmon, 1985). Their success in these areas, and others as well, is attributable to extracting knowledge from expert users and encoding this knowledge in a machine-readable format of objects encompassing rules and actions. Early Navy experiments in the ACCAT program were conducted using this technology for (C3I) planning (e.g., for air strikes). Current technology is using LISP-like artificial intelligence programming languages (McCarthy, 1962), with deviations for efficiency using the "C" programming language (Ritchie & Kernighan, 1978). A definite problem exists with the amount of data that can be accessed efficiently. Some of the newer systems now available plan for more efficient database management (PROLOG, 1988).

User Interface

Additionally, the OODB capability can be provided with a more user-friendly, natural-language access to database systems. This can range from menu selection, to ease DBMS access, to more natural English-language access.

With the emergence of a few truly grammar-based (as opposed to keyword-based) commercial natural language interfaces (NLIs) in recent years, natural language understanding (NLU) technology is finally emerging from the laboratory into the real world. As with many other emerging technologies, questions arise concerning the most productive applications of NLIs and the proper methods for evaluating the interfaces currently available. A study (Maslin & Sundheim, 1990) is currently underway at NOSC to evaluate the use of two commercial interfaces (Bolt, Berries, and Newman's "Parlance"; and Natural Language, Inc.'s "Natural Language" [formerly Data Talker]) and their configuration tools.

The two interfaces were evaluated in the following general areas:

- System/Architecture
- User Interface

- Development Environment
- DBMS Commands
- Customer Support
- Coverage
- Habitability

Unfortunately, from the results of this evaluation, these interfaces are apparently not yet ready for integration into real deployed systems, specifically because of their lack of linguistic coverage and a helpful user interface. More information can be found in Maslin and Sundheim's (Maslin & Sundheim, 1990) study of Natural Language Understanding (NLU) Systems.

9.0 DEFICIENT AREAS/CURRENT RESEARCH

The following discussion covers deficient areas in Navy requirements for the DBMS programmatic interface and how current research efforts in each area (if any) are provided in support of current and future standardization efforts for industry.

9.1 REAL/CRITICAL TIME PERFORMANCE

Deficiency

Commercial systems have traditionally not been driven by the same type of realtime/critical (UniForum, 1989) time concerns that drive military systems. In particular, most commercial systems do not have to meet hard deadlines which, if not made, could mean mission failure and loss of life. For realtime data, the issues of maintaining and managing realtime data consistency (data are not lost) while querying large databases (upwards of gigabytes in size in raw data) are tantamount, e.g., the ACDS and NTCS-A efforts referred to previously in this document. This is especially difficult where processing of simultaneous external asynchronous events is required. Industry now is becoming more aware of the issues facing realtime/critical time performance. Realtime demands are being met for robotics and process control systems for manufacturing. Even banks are beginning to appreciate the meaning of "realtime" to a customer standing in line waiting for an ATM machine. This increase in industry's awareness is starting to affect the systems being produced and, eventually, the standards that can result.

To ensure that data are processed in realtime, memory and other critical resources can be allocated and scheduled in close cooperation with the operating system. If memory is appropriately scheduled anywhere on the distributed database network, a transaction could run in realtime. No industry standards are being considered to resolve this problem and how the problem should be addressed in an open client/server network. Memory scheduling problems are being considered for a single processor by the Process Memory Locking feature in the POSIX Portable Realtime Operating System Interface Specification (IEEE 1003.4/UniForum, 1989), but not for multiple heterogeneous nodes on a network.

Realtime consistent updating of distributed databases currently is not available. Two-phase commit is still in its infancy; this is the method that allows data to be replicated or updated on multiple databases across a network in realtime. Two-phase commit will make sure that the replication session is completed at each node on the network, without interruption before the updates are committed to database memory. This prevents the problem of half-completed or interrupted sessions on some nodes that results in unsynchronized data. Presently, no standard exists for implementing two-phase commit, although the ISO is considering such proposals. No DBMS vendors currently support two-phase "realtime" commit.

Current Research

Research is being conducted in distributed realtime databases where a significant part of realtime data is highly perishable in the sense that it has value to the Navy mission only if used within time constraints, such as deadlines. From the realtime scheduling point of view, the primary problem introduced by sharing distributed data is the blocking caused by the locking or time stamp protocols for concurrency control that often cause unacceptable delays. However, concurrency control protocols are needed to ensure the consistency of the shared data (database) and the correctness of distributed computations (transactions). In the Distributed C2 Project at NOSC, experimentation is taking place with the Realtime Database (RTDB) environment (a transformed relational prototype realtime database). The RTDB was received from the University of Virginia (U of VA) and the Carnegie Mellon University (CMU) Advanced Realtime Technology (ART) operating system. The objective of the experiment is to evaluate the performance of the realtime environment, concentrating on the system's capabilities, limitations, time-driven schedulability, integrity, and predictability. For a description of the experiment, refer to Butterbrodt and Green (1990).

9.2 DATA CONSISTENCY

Deficiency

For static Navy C3I data, there is the issue of how to design the database system so that retrieval and execution performance do not suffer. For relational DBMSs, the technique of indexed columns—a non-SQL standard—is an extension that can be used to improve performance by allowing access to only a few columns of a relation. However, this is an ad hoc way to improve database query response. Any application requiring nonrealtime (NRT) large databases needs timely and consistent access to the data when these databases have few updates and/or realtime (RT) data for threat analysis and weapons deployment.

Such designs must assure that arriving data will be available, retain referential consistency with other data copies, and not be mistakenly lost during updates or while accessing large databases. The potential for processing concurrency must be available for determining the location and accessibility of the data wherever they reside. The designs must accommodate high data availability with responsive access and good potential for decision quality in accessing and identifying data. They must also allow for dynamically reallocating relation locations based on processor availability.

Current Research

A study (Small, 1990) is currently underway at NOSC that presents design options for a realtime distributed database system in support of Naval Tactical Command Systems-Afloat (NTCS-A) databases. It provides support to ensure that arriving data will be accessible and consistent with other available copies. The design consists of a directory (available at each node in the distributed system) that can be accessed by any application requiring realtime and nonrealtime data for threat analysis and warfare planning (refer to the discussion about distributed systems in Section 9.4 of Chapter 9 in this document). All accesses will be provided by SQL relational syntax commands. The potential for processing concurrency can be available as can a methodology for determining the location and accessibility of the data wherever the data reside in the system. The design provides high data availability with responsive access and good potential for decision quality in identifying target data. The implementation currently supports only fixed processor location of relations and directory presence only on the node responsible for application processing actions. Smooth growth in future years is provided for (1) allocating more processors and (2) recovery from processor failure or processor overload.

The database will then be enriched with respect to more data, larger databases, deliberate insertion of errors, and transactions to encompass rules for investigation of decision reversal concerning who is accessing or allowed to access the data. Other experiments in forcing data consistency will be selected. In all cases, investigations of the adequacy of the directory module will be considered as will different methodologies for data distribution that will maximize performance and data quality.

9.3 HETEROGENEOUS DISTRIBUTED DBMS' INTEROPERABILITY

Deficiency

A clear trend in industry is toward both heterogeneity and distribution. Note, however, that no industry interface standardization efforts are specifically aimed at distributed DBMSs. See the reference to "The Promise of Network Databases" (Davis, 1990) that states a definite need for "secure, distributed databases with realtime updating and standardized query languages." Some work is being performed in network management and with operating systems, particularly at the ISO level, that may be relevant to the problem. However, these efforts are unlikely to adequately address all DBMS/operating system/network interface problems.

To solve the problem of communicating uniformly among disparate, very large heterogeneous and distributed databases, the Navy is considering the use of SQL for accessing and updating data in an open client/server database system (Small, 1990). Such a system can ensure that arriving data will be available at any time and be consistent with other copies in representation and time. The client processor enables the analyst to communicate with his application. The operating system server brokers can then provide the required database accesses and updates for the application. With this split in processing capabilities, the user can be assured of a timely response to the requests. Industry also is considering such architectures. The Navy, in selecting an industry open client/server system, must carefully avoid getting caught in a proprietary network arrangement. This can be avoided by promoting a standard for such architectures. However, this problem still exists for transferring large files in a timely manner between heterogeneous databases. Industry is investigating the problem of addressing heterogeneous dissimilar database systems, but primarily in batch systems applications and not interactively. Again, in this latter area, no standardization is being considered.

Current Research

The access to remote data could conform with the emerging OSI standard called Remote Database Access (RDA) (ISO 9579-1, 9579-2). This standard is a medium through which such a client/server architecture can evolve. The SQL Access Group expects to (1) build several prototype clients and several prototype servers (connected to existing commercial DBMS products) and (2) demonstrate the interoperability of those clients among the servers using the RDA protocols over an OSI network stack. The RDA specification is being developed within ISO before the implementation phase. This prototype will serve to validate the RDA model and help detect shortcomings early so corrections can be incorporated into the RDA standard before it is finalized. At this writing, the standard is at second-draft stage of the proposal.

Because the prototype will use existing DBMS products, the servers will typically convert the RDA protocol into dynamic SQL statements and return result values using the RDA protocol. Capabilities available to the client are thus necessarily limited to those that are generally implemented in commercial products. This is why the X/Open SQL specification was selected as a starting point for the embedded language definition. As a consequence, initially, these capabilities will demonstrate a single client/single server connection (not requiring two-phase commit in the DBMS), although the general problem of multiple database access within a single transaction is being considered in the specification. In follow-on phases of the specification, call-level application interfaces, among other things, will be added to increase interoperability. This work definitely warrants monitoring by the DBWG.

Even with the use of similar relational database systems, all query languages among relational DBMSs are not alike. Most vendor SQL implementations vary in SQL support and are not compatible, even though they use ANSI SQL as their core (refer to the Heterogeneous section [7.2] on Navy Requirements for DBMS Interface). To date, no solutions exist for this problem. In response to this, the SQL Access Group is a vendor consortium that was formed to establish standard interfaces and protocols to allow users to access data from different vendors' relational DBMSs. The objective of the SQL Access Group is to solve the database interoperability problem by persuading major DBMS vendors to agree upon and implement a standard SQL server interface so that one vendor's SQL application can access another vendor's database. The interface being developed by the SQL Access Group is based on existing standards, such as ISO/RDA and ISO/SQL. The group is committed to working with the standards organizations and will openly publish their technical specification. NIST (National Institute of Standards and Technology) has recently approved the "core" SQL Standard; refer to ANSI X3.135-1986 (ANSI, 1986).

This specification will take off from that standard and will include a standard Application Programming Interface (API) and a standard Formats and Protocols (FAP). The API work group is working from the ANSI/ISO SQL specification as reflected by the X/Open Data Management Portability Guide. The objective is to provide users with a language manual for embedded SQL that will be portable and interoperable when used in portable applications. The FAP work group is working from the ANSI/ISO Generic RDA and SQL Specification documents, with any additional items carried as a "Differences Document." The FAP work group works closely with the ANSI X3H2.1 organization (ISO 9579-1 and 9579-2).

9.4 DISTRIBUTED SYSTEMS

Deficiency

Currently, distributed operations cannot be performed where dissimilar database structures and large files reside on multiple network nodes, without using proprietary software. An example of proprietary vendor network software would be Oracle's SQL*Net product. Current research could provide methods for using SQL commands to access Navy relational databases that may be distributed over multiple locations (i.e., a distributed query across a network). These methods should allow a local area network (LAN) user to (1) issue a SQL SELECT statement that addresses multiple tables residing on multiple remote computers and (2) return a final result without using proprietary network software.

Current Research

Directory Services. A study (Small, 1990) is currently underway at NOSC that presents design options based on a directory system for a realtime distributed database system in support of Naval Tactical Command Systems-Afloat (NTCS-A) databases. (See the discussion under the Deficient Areas chapter [9.0] on Data Consistency [9.2].) The design consists of a directory available at each node in the distributed system that can be accessed by any application requiring realtime and nonrealtime data for threat analysis and warfare planning. Among the issues of concern in the continuation of this study will be how this design may be influenced by the new OSI X.500 Directory Services standard, and the use of very dissimilar database structures and large files.

The OSI X.500 Directory Services Standard will enable users to send messages and files, for example, to remote users without having to know or manually look up the location of the recipient. The standard will provide a specialized distributed database for OSI applications. It will contain information about objects and then provide a structured mechanism for accessing that information. The information is collectively known as the Directory Information Base (DIB). Directory Services are intended to aid in information distribution and retrieval and to aid in network management. Directory Services are also intended to provide user-friendly naming. This permits a user of the Directory (not necessarily a person) to specify an object's name and then to retrieve additional addressing information. There are two important uses for this. The first is "name to presentation address" mapping for OSI applications and second, "name to electronic mail address" mapping for use with message handling systems. Remember that the Directory is not intended to be a general purpose database. The Directory will be used mainly for queries (reads from the Directory) rather than for updates (writes to the Directory); and a hierarchical, rather than relational, architecture will be used for naming. The Directory is designed for large scale and long-lived networks. Because of the large scale that the Directory must address, and the inherent delegation of authority needed for such a vast undertaking, the methods used for identifying objects have been optimized primarily to facilitate allocation. This explains why names, the primary method for identifying an object, are hierarchical rather than relational.

Directory Services became an international standard (IS) in December 1988 (ISO IS 9594). The National Institute of Standards and Technology, Gaithersburg, Maryland, recently received funding to create a government-wide database directory based on the OSI X.500 Standard. NIST and the General Services Administration are working together on the X.500 effort. The X.500 implementation will be constructed on the ISO Development Environment (ISODE) and run on a SunOS Unix platform. Implementation of a government-wide X.500 directory should allow the government to establish the credentials needed to put X.500 into Version 3 of the U.S. Government Open Systems Interconnection (OSI) Profile (GOSIP).

Distributed Query Optimization. The Distributed Query Processor (DQP) is the name of an engine dedicated to working out distributed query access for DDBMS. Under the auspices of NOSC, Code 413, research is continuing on a design (Mollet, 1990) that includes optimization techniques to reduce communication costs, wall clock time, CPU time, memory, and disk utilization. The DQP executes the following steps: (1) gets the query from the user, (2) validates the query, (3) segments the query into individual "subqueries" appropriate to each remote database, (4) sends each "subquery" to the appropriate hosts for processing and awaits return of the resultant table(s), (5) interrogates tables returned by the remote hosts and produces a final report, and (6) returns the result to the user. The proposed solution is simple and clean, because the DQP is built over the ISO/OSI seven-layer stack as implemented in the ISODE software. This approach conforms to National, International, and U.S. Government (GOSIP) standards. Also, since the DQP is a non-proprietary product being developed by NOSC, absolute control is provided over its operations, thereby eliminating many other associated problems.

To summarize, the DQP can take the SQL query, break it into subqueries, and concurrently make requests to the appropriate remote database(s). In turn, the DQP can read the resultant tables, join the subqueries into one answer, and produce a report for that end user. This technique will reduce communications costs significantly which is very important over low-bandwidth tactical networks. Commercial DDBMS query optimization techniques are still maturing.

9.5 LANGUAGE BINDING

Deficiency

The automatic DBMS interface should be defined independently from any particular programming language. To date, every relational database system has its own programmatic mechanism for accessing its services. Appropriate means for such access should be provided.

Current Research

Ada's support for automatic access to database systems is increasing. One important indicator is that POSIX has another subgroup, P1003.5 (POSIX 1003.5/UniForum 1989/NGCR OSSWG Report), that is working on an Ada binding for the interfaces. This will also affect the DBMS community. Another is that in February 1989, an Ada/SQL Ada-Database Management System (DBMS) language interface was developed for use by the WIS (WWMCS Information System) Joint Program Office (Institute for Defense Analyses Documents D-574 and 575, 1989). This interface to SQL was designed as an extension to SQL, titled Ada/SQL. It adds Ada's type declaration and checking capabilities to SQL. The schema definition language for Ada is provided through Ada/SQL/DDL, and data manipulation is through an extension titled Ada/SQL/DML. Using packages such as these, a consistent mechanism could be provided for data definition and access.

However, Ada is not yet fully integrated into all of industry's interests. Since many industry standardization efforts are influenced by UNIX, which is tightly coupled with the programming language "C," the resulting standards are most often expressed as "C" bindings, not Ada ones. The Department of Defense (DoD) also recognizes FORTRAN and COBOL as approved standard languages that are used many times. Refer to previous sections on DBMS Utilization in Strategic Systems and in Intelligence Systems. To smoothly preserve existing DBMS utilizations, those bindings should be provided within the DBMS NGCR Interface. The ADA 9X committee is also investigating similar realtime issues (Ada 9X Report August 1990).

9.6 MULTILEVEL SECURITY

Deficiency

Like realtime, security traditionally has not been as great a concern to industry as to the military. The military is concerned about issues of confidentiality, data integrity, non-repudiation, proof of origin, and submission. But also like realtime, these security issues are now becoming increasingly important to systems in use for nuclear control and banking. Security implications are not well understood in such systems, but there are guidelines, and the requirement is real. Such a system should ensure that users cleared at different security levels can access and share a database without violating security. Industry's recognition of this is demonstrated by yet another POSIX subgroup, P1003.6, working on security enhancements for interfaces (POSIX 1003.6/UniForum 1989/NGCR OSSWG Report).

Two installed, procurable database systems that we know of have been certified and accredited to run controlled/limited multilevel security. These are the M204 system, now being installed as the WWMCCS standard system; and the Honeywell Integrated Data Store 2, that has formed the basis for the Navy's World Wide Data Management System, WWDMS. None of these provides an open client/server system. In all instances, the system requires specialized machine and/or operating system support.

Referential integrity, i.e., maintenance of consistency when changing related data in other tables, must be provided whenever using SQL commands such as insert, delete, and update. The use of view trigger mechanisms may provide the basis for acceptable Discretionary Access Control (DAC) in a DBMS. The combination of such access control with management of data integrity using data validation and consistency constraints could provide a reasonable level of security for user access to data. This does not say that the system is tamper proof or has met all requirements for a secure system. Providing such access control does provide a starting place for multilevel security. Multilevel relational systems are now under development that will give different views of data at different security levels. The mechanisms use polyinstantiation to allow two different tuples to exist with the same primary key. Maintaining and managing items in such a system can be costly, particularly if that requirement must be met for each row of a table or file record at different security levels. But regardless of such potential cost, new systems under development that operate in a multilevel secure environment (whether they are object oriented, knowledge base, realtime, distributed database systems, etc.) must provide for multilevel security at the beginning of their development. Research is currently underway in inference and aggregation, using multilevel secure systems, SQL extensions, integrity policies, and automated auditing techniques, all to support more secure systems.

Current Research

As a part of an Air Force project, Oracle Corporation plans to enhance its DBMS' security level to A1 and port it to run under Gemini computer's GEMSOS secure operating system. This enhancement will be based on a model built for the Secure Data Views (SeaView) project, a program for developing a secure relational DBMS for Defense Department applications. In addition to a full implementation of current ANSI SQL security provisions, Oracle will provide a full security auditing facility designed to meet existing Orange Book criteria. Oracle's contract provides for developing two types of security features, discretionary and mandatory, to meet the needs of commercial users as well as the DoD and intelligence communities. Anticipated new discretionary features will include group-level access controls, improvements to Oracle's auditing facility, and additional administration roles. The new mandatory security capabilities will provide support for multilevel separation of classified data of different compartments and categories. To implement these capabilities, Oracle will use a trusted computing base to enable its DBMS to run on secure operating systems available from Digital Equipment Corporation (DEC), IBM, and several Unix system vendors. Specially designed secure operating systems will also be used; e.g., GEMSOS from Gemini Computers. Oracle will work with Gemini Computers and SRI International on the project. Funding is from the Air Force and the Rome Air Development Center at Griffiss Air Force Base, NY.

Sybase has released their commercially available multilevel secure DBMS. The company's Secure SQL Server is designed to meet NCSC's B1 requirements (NCSC, 1990). DEC has teamed with Sybase to sell Sybase Secure SQL Server and Toolset for use on VAX Ultrix systems. The Secure Server (as of this writing) runs in Ultrix, SEVMS, and SUN MLS environments, as do client tools for DEC's VMS and Sun Microsystems' platforms.

In addition, the commercial trusted DBMS' Atlantic Research Corporation's TRUDATA, Informix, Infosystems Technology, Inc.'s, Trusted Rubix, and Teradata are under consideration for certification at the B1 level.

Informix is intended to run on the HP UX 804 (HP RISC machine targeted at B1), as well as the already evaluated AT&T System V MLS. TRUDATA merges trusted code and encryption technology and runs on a combination of a Britton Lee 3B2 and AT&T MLS.

Teradata has developed a database machine targeted at the B1 level. This work, a research prototype funded by NCSC, was delivered to NCSC in February 1991. Because the system is a database machine, there is no underlying database machine.

Infosystems Technology Incorporated is developing a Trusted Database Management System (TDBMS) prototype targeted at the B2 level. The TDBMS is known as trusted Rubix and is intended to run on AT&T's B2-targeted operating system. The work is being funded by RADC.

Although Sybase has released their multilevel secure product and the other vendors listed are under consideration for certification, questions still are being raised about NCSC's evaluation process. NCSC has not yet issued formal DBMS evaluation guidelines, but has issued a third draft of a Trusted Database Interpretation (TDI) for its Orange Book. That document, officially the Trusted Computer Security Evaluation Criteria, describes only secure operating systems (NCSC, 1990).

9.7 FAULT TOLERANCE

Deficiency

Fault tolerance and reliability are clearly recognized industry concerns. Reliability issues are much the same in both industry and military systems. However, fault-tolerance issues must be treated differently in military systems, since they must remain on-line at the height of mission critical situations, be concerned with dynamic reconfiguration, and not suffer critical data loss when reconfiguring. Again, the prime issue is performance for mission completion. The lion's share of commercial systems is not concerned with the costs and anomalies of on-line dynamic reconfiguration. These systems must perform background (albeit on-line) trailing and archiving of data. Then, during failures, most of the reconfiguration can be manual or moderately automated, and transaction processing with possible input data loss can be tolerated better.

Current Research

Standardization efforts in these areas are unknown, and any work done here is application and requirement specific. The specific areas of "commit" and "rollback" should be studied more for incorporation into systems for more reliability and integrity of data kept between and among inter- and intra-databases. This concept may degrade performance and is as much a network and operating system issue as it is a DBMS issue. The research on Data Consistency (Small, 1990) discussed in Deficient Areas/Current Research (Chapter 9.0) of this document could have an impact if integrated into a total concept for fault tolerance.

9.8 DBMS LOGISTICS SUPPORT

Deficiency

A major issue is keeping current with Navy data via updates and preserving such data when the database system changes. Referential integrity maintenance is vital to such preservation, just as it is for fault tolerance and recovery, and multilevel security. In providing such changes, there must be assurance that data to be entered in the database system will not become contaminated and that data do not become lost or out-of-date.

Research

We believe nonrealtime techniques exist that can be used to manage data updates using commercial standard DBMS'. The problem becomes somewhat more difficult if the data are realtime (see the aforementioned section on Data Consistency), if they must be merged with similar types of data stored using a different DBMS, or if they must be converted to a different DBMS. To date, we know of no particular solutions to these problems. Today, the Navy is supporting NWTDB's life cycle by doing all maintenance in one single organizational structure.

9.9 NUCLEAR THREAT IMPACT

Deficiency

Industry pays little attention to the problem of nuclear survivability. We must address the issue of how to ensure the integrity of the actual original executable program code and stored data. All magnetic media, whether on-line at the time of the nuclear incident, or off-line and "safely" stored for use in subsequent manual recovery and cold (re)start operations, is subject to damage. If the executable program code and stored data do not have a known integrity, data recovery is of little or no value. Manual cold (re)starting of a carrier-sized platform is no trivial situation and can involve time-consuming operations similar to those involved in data restoration for fault tolerance and recovery. The mass storage problem probably can be solved using CD ROM backup of program and more static data files. This could make the software at least as survivable as the computer hardware. This issue will be handled within the whole NGCR architecture.

Current Research

See Fred Warnock's (NADC) paper on this topic (Warnock, 1990).

10.0 EMERGING STANDARDS' IMPACT ON DBMS INTERFACE

This chapter addresses several emerging standards and their impact on the NGCR DBMSIF. The standards work mentioned include the realtime features of the POSIX Portable Realtime Operating System Interface for Computer Environments (IEEE 1003.4/UniForum 1989/NGCR OSSWG Report) standard, the open system features of the U.S. Government Open Systems Interconnection (OSI) Profile (GOSIP), Implementation Agreements for Open Systems Interconnection Protocols, NBSIR 86-3385-5 (NBS 1), the SAFENET II (SAFENET, 1990) networking standards, and the "Orange Book" Trusted Computer Base (TCB) (NCSC, 1990) provisions for multilevel security.

10.1 POSIX

POSIX refers to a standard application interface for portable operating systems that was promulgated in August 1988 as Federal Information Processing Standard 151. Its importance lies in the fact that it is the first attempt to specify a nonproprietary common set of program calls and command line interfaces for an operating system. In the future, many operating systems are expected to offer compliant interfaces and subroutine libraries.

The baseline operating system interface chosen for the NGCR Operating System Interface-4 (OSIF) is POSIX. It will support all resources or provide management for them. Application examples include scheduling the use of internal memory, network facilities required, and external memory (other memory or processing elements) resources, such as disk and tape drives. To support NGCR DBMSIF requirements, the NGCR OSIF must accommodate a variety of scheduling approaches, along with the ability to influence them. An example of this is using priorities to influence the designation of which processes are to be scheduled for controlling memory or for controlling network access. The interface must support or provide resource status monitoring so the DBMS can better determine where data access can best be accomplished. The interface must be designed so that realtime system developers can configure the operating system and DBMS implementations to best suit their needs.

The POSIX FIPS and GOSIP FIPS are complementary, and their effect is expected to be synergistic. The POSIX standard will be used to provide a favorable software development and execution environment for many applications, including data access using OSI protocols. The GOSIP standard will be one of those used to achieve interoperable data communications between computer systems.

10.2 GOSIP

The data management issues in distributed systems will be among the more difficult issues to resolve. To solve the issue of uniformly communicating among disparate, heterogeneous, and distributed databases, the Navy is considering the use of SQL for accessing and updating data in an open client/server database system. The client/server model is one in which two independent processes communicate with each other, one (called a server) supplying a service to the other (called a client). The client is the requester and enables the analysts to communicate to their applications. The operating system server brokers can then provide the required database accesses and updates for the application. With this split in processing capabilities, the users can be assured of timely responses to their requests. Industry also is considering such architectures. The open-system protocols of the GOSIP communications/network, that could partly be based on the SQL open-client/server architecture, can guard against their use being caught in a proprietary arrangement.

However, no definition exists in GOSIP regarding how the DBMS can fit in as an application for layer 7. Initial application layer protocols of OSI referenced by GOSIP are listed below. (Supporting protocols at layer 6 and below are assumed and shown in the diagram of figure 2.)

- File Transfer, Access, and Management (FTAM) [NBS 1;ISO 16-19], which address access and movement of information files among network users.
- Message Handling System (MHS or X.400) [NBS 1; CCITT 2-9,14], which addresses electronic mail or messaging between network users.

The following is an explanation quoted directly from the GOSIP Version 1.0 Standard as it existed in January 1988 (GOSIP, 1988):

> "An open system is a system capable of communicating with other open systems by virtue of implementing common international standard protocols. End systems and intermediate systems are open systems. However, an open system may not be accessible by all other open systems. This isolation may be provided by physical separation or by technical capabilities based upon computer and communications security.
>
> GOSIP must be complete because open systems procured according to it must interoperate and must provide service generally useful for government computer networking applications. Since this specification is one of open systems, the secondary sources include specifications that are international standards or are advancing to become international standards. They are included in GOSIP to help satisfy the criterion of completeness, and, thus, utility.
>
> The principal thrust of OSI is to provide interworking of distributed applications using heterogeneous, multivendor systems. Modern implementations of OSI products may perform adequately for most government applications. GOSIP does not cite performance criteria." (Jackson, 1990)

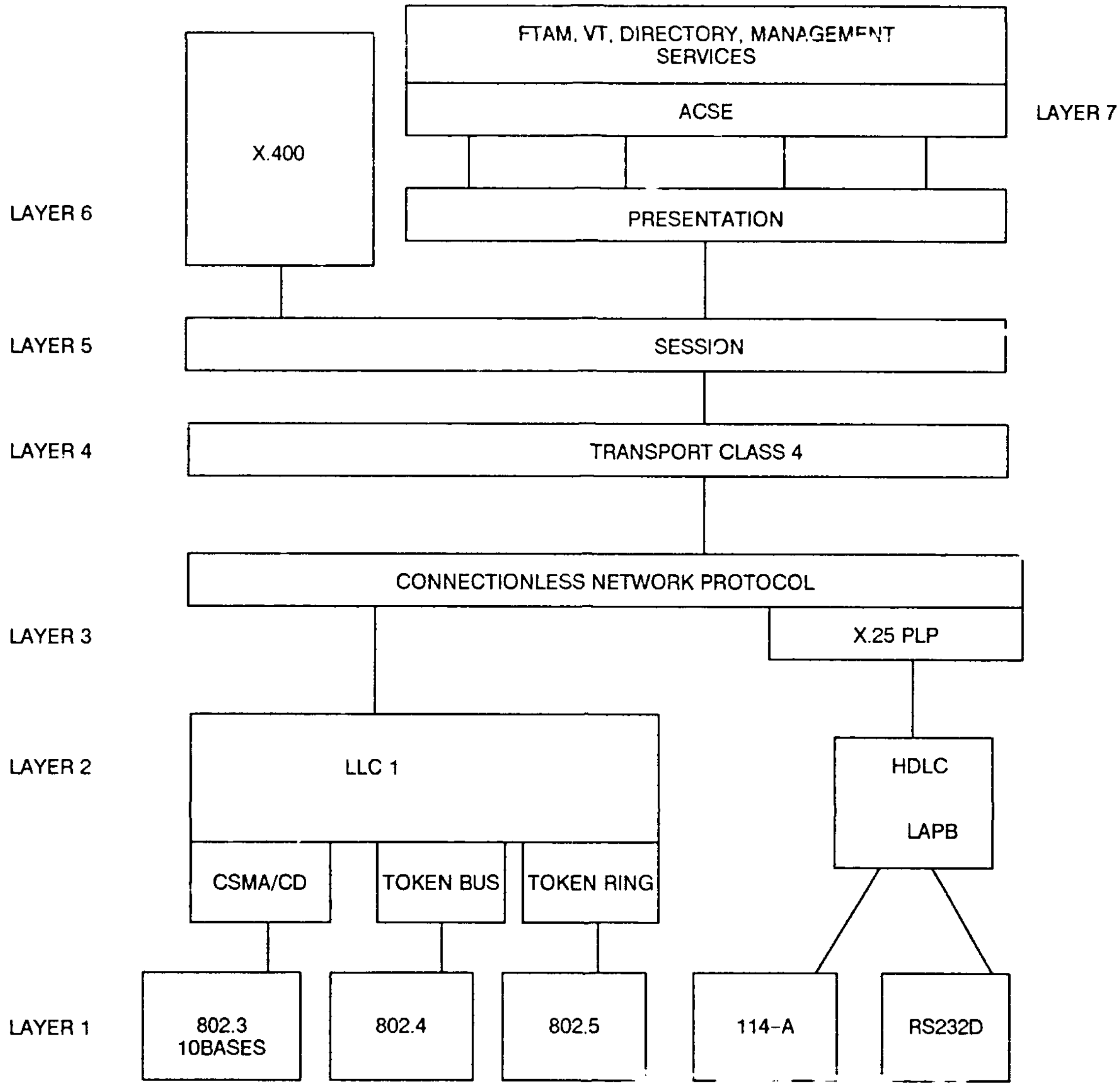

Figure 2. Government OSI profile architecture.

Version 1.0 of the GOSIP profile is comprised of the X.400 Message Handling System; File Transfer, Access, and Management; Transport Class 4; Transport Class 0; the Connectionless Network Protocol; X.25; and the Institute of Electrical and Electronics Engineers 802.3, 802.4, and 802.5 local area network standards. The timetable for GOSIP is as follows: Version 1.0—Effective 15 August 90; Version 2.0—Draft out September 1990, effective 18 months later; Version 3.0—later in the 1990s (Jackson, 1990).

At this writing, the services listed below are not required in the GOSIP Version 1.0, but some form of these services is required by an operating system for reasonable resource management. For the DBMSIF to take advantage of GOSIP, the following services must be implemented.

- Directory Services (X.500): X.500 is not scheduled to be part of GOSIP until Version 3.0. See the paragraph on Directory Services under Deficient Areas for Distributed Systems (Section 9.2).

- Association Control Service Element (ACSE): ACSE is responsible for association, establishment, and release, each of which is required by Directory Services. An example could be establishing an association of a logical connection between a SQL query client and a SQL query server.

- Remote Database Access (RDA): RDA is an emerging standard intended to support internetworking between an application program in one open system and a DBMS in a remote open system. The standard governs only the communications aspects of such interworking and interface with the application layer (layer 7). The GOSIP and RDA applications can be complementary.

- Heterogeneous Database Access: This problem requires resolution. A programmatic interface must be developed for each heterogeneous DB. For example, if the user wanted to access Oracle and Sybase data on the LAN, a programmatic interface would have to be developed for Oracle and Sybase. Computer Corporation of America's MULTIBASE System software architecture of October 1982 could be part of a model for retrieving data from pre-existing heterogeneous distributed databases (Computer Corporation of America, 1982). The Naval Postgraduate School is also conducting research in interoperability and integration in heterogeneous database environments. A recent paper describes two levels of data access and sharing in such a database environment (Kamel, 1990).

In the Distributed C2 Project at NOSC, experimentation has been evolving in exercising the ISO Development Environment (layers 5, 6, and 7) on top of TCP/IP and remote DB access using Oracle, Informix, and SYBASE (Butterbrodt, 1990). Refer to figure 3 for the functional configuration of the distributed database architecture.

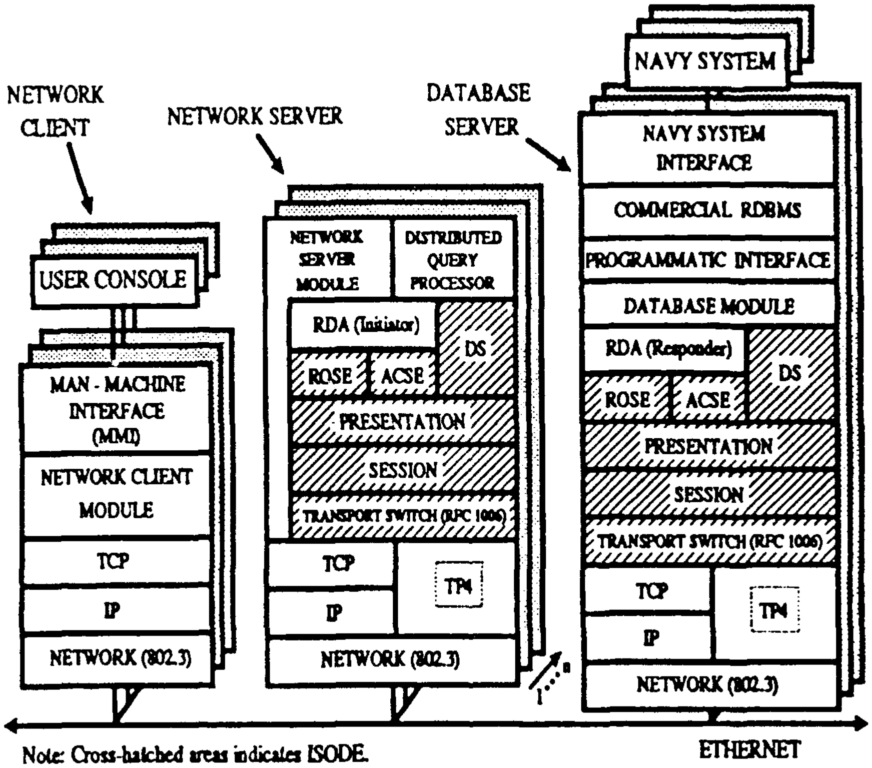

Figure 3. Functional configuration of distributed database architecture.

Figure 3 shows the functional configuration of the three major components: the Network Client, the Network Server, and the Database Server. The three components can be briefly described as follows: One, at the Network Client, a user can input an SQL SELECT statement using the man-machine interface. The Network Client module will send the query via TCP/IP over the Ethernet network to the Network Server. Two, the Network Server module will break the query into "subqueries" (via the DQP), next interacting with the RDA Initiator. It will then search for the remote database(s), via the OSI protocol stack, and access its corresponding remote RDA Responder(s). Three, on the remote Database Server, the RDA Responder(s) interfaces with the programmatic interface(s) to the commercial DBMS(s) and retrieves the relation(s) that satisfies the subquery. The relation(s) is returned to the Network Server, and the DQP consolidates. Four, this returns the answer to the user at the Network Client.

10.3 SAFENET

The Navy is developing requirements for Local Area Networks (LANs) to support mission-critical computer resources. The Navy must have a communications architecture for combatant ships that can handle massive host-to-host file transfers, extensive scientific and engineering applications, and other Navy large-scale requirements. Current LAN technologies, where workstations are generating more graphics and larger file sizes, are increasing and soon will not be able to handle the data volume within their design throughput envelopes. In the next 2 to 3 years, as workstations resident on LANs are used to generate more graphics and larger file sizes, the need to use fiber optics will expand. The DBMSIF should interoperate with the Survivable Adaptable Fiber Optic Embedded NETwork (SAFENET) standard (SAFENET, 1990), because the network will provide predictability and realtime performance when required to execute massive data transfers.

Based on these requirements, SAFENET will provide one of the logical link control (LLC) data link options, the token ring, for layer 2 of the GOSIP architecture. This option is based on the Fiber Distributed Data Interface (FDDI) standard as defined by the ANSI X3T9.5 Committee. SAFENET uses all fiber optic cabling and has dual-redundant counter-rotating token rings with a 100 Mbps data rate. SAFENET specifies three protocol suites for interoperability: (1) an OSI suite for interoperability of heterogeneous computer systems, (2) a lightweight protocol suite to support realtime applications, and (3) a combination suite that includes both (1) and (2). (See figure 4.)

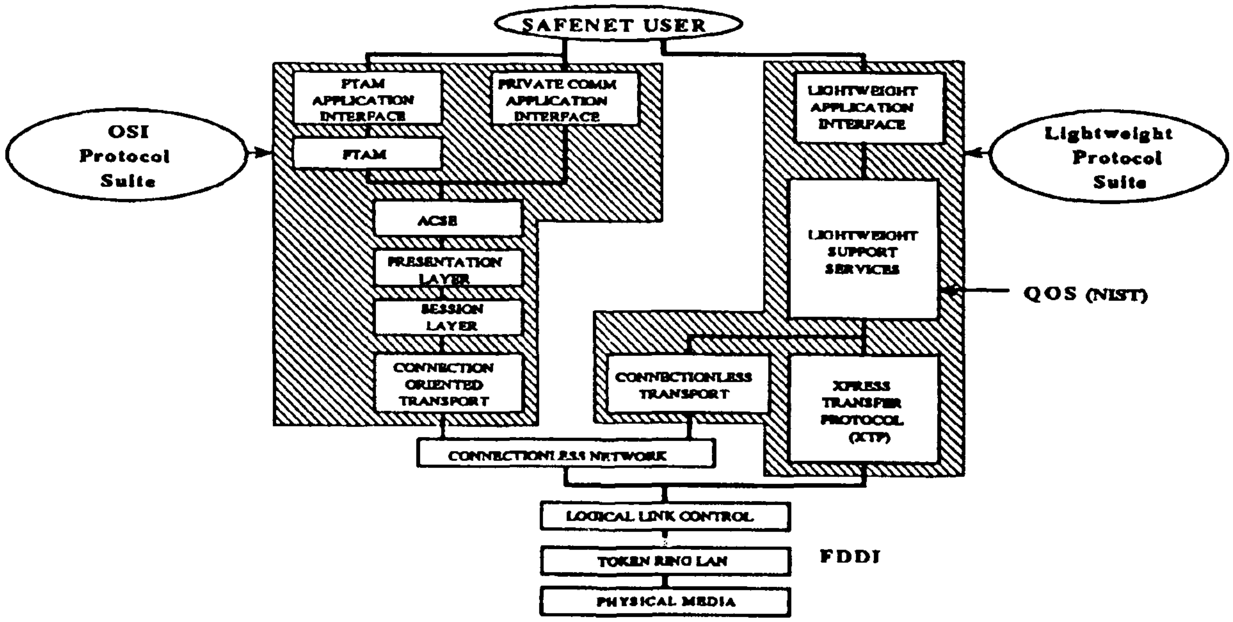

Figure 4. SAFENET protocol suites.

For realtime performance, SAFENET's Lightweight Protocol could be used as a direct connection between GOSIP's Application layer 7 and the support services provided by GOSIP layer 2's LLC. Within the lightweight suite, an information packet can take two paths. One path uses a connectionless transport layer that delivers packets with a best effort, no acknowledgement scheme; much like a letter in the mail. The other path uses the Xpress Transfer Protocol (Saunders & Weaver, 1990) that is a high speed, reliable scheme that assures delivery of error-free packets in the correct order. This is an essential service for mission-critical applications that depend on delivering data accurately with low latency (end-to-end delay).

10.4 MULTILEVEL SECURITY

Determining whether the DBMS or the underlying operating system provides the access mediation is a design decision. Currently, only Oracle has placed all of access control mediation in the operating system. Due to timing and performance considerations, as well as from a security point of view, the database must interface directly with memory management software, device driver software, and the system scheduler. From the "Orange Book" (NCSC, 1990) Trusted Computing Base (TCB) point of view, the DBMSIF would interface with a security reference monitor that could provide a subset M of the TCB. This subset will exist below the interface standard, transparent to applications, and will mediate every access to a set of objects (O) by subjects (S). M must be tamper resistant and small enough to be subject to analysis and testing. The "completeness" of M must be assured. From the context of the NGCR OS, a subset of POSIX could provide the M (POSIX 1006.3/UniForum 1989/NGCR OSSWG Report). The access control policies (P) of POSIX's M must provide at least

1. Assurance that there is no violation or penetration of protected memory space by unauthorized users (including log-on asssurance from the OS point of view).

2. From the multitasking point of view, assurance that there is no mix-up between tasks, and no invasion of code or memory used between tasks.

11.0 DBMS CONFORMANCE TESTING AND PROTOTYPING

During the demonstration and validation phase of the NGCR program, the complexity of the interface between various alternative, network, operating system, and DBMS candidates should be tested in an environment using realtime simulated threat environment data with hard deadline scheduling requirements. A main issue is how much sensitivity there is to various alternatives for types of hardware (i.e., nets, computer processor, and peripherals) utilized and the predictability of the functionality for NGCR system alternatives. Later testing in a more realistic operational environment can provide a better estimate of cost. In all cases, a nonintrusive mechanism for timing and testing is required.

Because of the lack of maturity in open systems architecture for heterogeneous database systems, coupled with realtime systems and maintenance of consistent data while querying large databases, some very important implications exist for the prototypes of requirements for the NGCR interfaces. First, implementations of the DBMSIF probably will not exist when the standard is published, if there is no NGCR prototype. This is because the resulting standard is likely to be an amalgam of multiple efforts that have not previously been integrated. Therefore, the NGCR program will be responsible for demonstrating the viability and implementability of the standard through prototypes. Second, the DBMS and the operating system, perhaps more than any other NGCR components, must interact closely with every other NGCR standard. Because of this, the program office must assure all of its potential users that these standards can be used effectively in concert with all the other members of the NGCR suite of standards. This only can be demonstrated by the development of NGCR-wide prototypes that investigate and verify the ability of all the standards to be used together to develop viable systems. The development of prototypes can also help to accelerate the development of commercial implementations. If the standard and the prototypes are developed properly, other vendors should be able to use many instances of prototype code, at least initially, to fill out their implementations. This will save development time and allow more time for vendors to concentrate on specifically unique aspects of their implementations.

A major concern in developing a new embedded computer system for the Navy—actually a temptation—is to build something brand new, requiring a new computer, a new operating system, a new computer language, and a new DBMS to interface with the new operating system and computer. Hard Navy experience has shown the cost of taking that alternative—i.e., making a modern NTDS by building ACDS with AN/UYK-43s instead of AN/UYK-7s and earlier 30-bit word AN/USQ-20s, as an example. The Ada computer language was started to facilitate building and transporting new systems from one machine to another. But early on in the Ada development, languages were acknowledged as having to interface with other technologies—i.e., operating systems, distributed network systems, database systems—most of which have the realtime requirement. Doing rapid prototypes of a concept by using commercially available equipment has proved successful and less expensive. The hazard of doing this is that sometimes the Naval commanders want to keep the equipment. Of course this is a pleasant hazard until the "ilities" catch up! Many of the concepts are difficult to handle and require experimentation—particularly in the world of realtime open system architecture for distributed database systems. Many experiments can be performed using simulated unclassified data with commercially available equipment that contains built-in accessible clocks. These experiments can then be used in developing standards to which industry can build. (See the following chapter in this document on DBMS METRICS.) One of the more successful of such experiments resulted in building the commercially available SAFENET LAN (SAFENET, 1990) hardware in support of ISO layers 1-3. The cost of this is time and country-wide coordination with interested vendors. The payoff can be supportable and responsive deployed (C3I) systems.

12.0 DBMS METRICS

The TP1 (Transaction Processing Standard 1) and the DeWitt Benchmark (DeWitt, 1984) based on the CREDIT/DEBIT transaction banking model are sets of executable metrics used for judging database performance in the commercial world. This method was used successfully on tests for performance measurements using ORACLE Version 6 to support NTDS track data fill. (Butterbrodt, 1988a, 1988b, 1988c). One of the main issues is how to show predictable performance over a network of heterogeneous computing elements as a function of time in a realtime world.

13.0 NGCR POLICY

NGCR policy is to adopt existing commercial standards whenever possible. The world of DBMSIF-related standards is quite bewildering. That is, a number of standards and would-be standards seem to be applicable; but, clearly, no common vision coordinates them—as the OSI reference model does for the world of LANs. This makes it extraordinarily difficult to determine (1) which standards might be adopted together to achieve some goal or (2) where there are holes or gaps where no standardization activity has started. A critical step in establishing the DBMSIF interface standards will first be to establish a reference model that can help to bring some sense out of the chaos and then to unravel the maze of efforts and put them into some context with respect to this model.

Also important is for NGCR to carefully consider what the Navy's policies should be with respect to the mandate of the adopted DBMSIF interface standards. A "carrot-and-stick" approach (as opposed to a strictly "stick" one) would undoubtedly be most effective in an area such as the DBMSIF where a great deal of change is happening, with very little maturity of any current products or efforts.

14.0 APPROACH TO DEVELOPMENT OF DBWG

In FY 91, plans are being made to form the FY-92 working group. Work will begin on the preliminary drafts of the DBMSIF Working Charter, Available Technology Report, and Requirements Document. The DBMSIF working group will use these draft documents when it forms in FY 92.

The primary objective of the DBWG will be to develop a set of interface standards for the database management system environment. In support of this objective, a variety of accompanying documents must be produced, including at least the following:

- Operational Concept/Reference Model
- Requirements (with rationale) Document
- Rationale for the Set of Interface Standards
- User Guide and Implementer Guide

The capability to demonstrate the viability of the proposed standard through prototype implementations will also be a critical requirement for a successful effort. As the NGCR budget currently stands, only seed money is planned for such an effort, so it must be achieved through cooperation with other projects and cooperative use of available resources.

The DBWG should have primary responsibility for all decisions made concerning the DBMSIF standard, specification, and accompanying products. It should be structured analogously to the existing NGCR working groups, with a Navy Chairman and Co-Chairman, and a mixture of government, university, and industry participants. Meetings should be held at least quarterly, possibly supplemented by more frequent meetings of individual subgroups.

Before the DBWG is first convened, the Navy laboratories, under the leadership of NOSC, will do further planning. This planning should be further developed and elaborated on, based on the suggestions presented here for organization, issues, and products. The first DBWG meeting should be attended by only government personnel. This is to ensure coherence and direction of the government objectives and requirements prior to exposing them to the general community. Such an initial government meeting can be pursued in parallel with the solicitation of initial information from industry and universities.

Government participants should be solicited from at least each of the appropriate activities of the Navy laboratories. Other sources of relevant expertise should also be investigated and tapped, if possible, including Navy development and testing activities; and other federal agencies, such as DARPA (Defense Advanced Research Projects Agency), NASA (National Aeronautics and Space Agency), and NIST.

Industry and university participants will be solicited both from known sources and through open solicitations, such as in the Commerce Business Daily.

The DBWG should assume both that the government does not have sufficient qualified personnel by itself to successfully complete this project and that volunteers (whether from the government, academia, or industry) cannot be expected to be sufficiently regular or dependable. Thus, plans should be made to have two kinds of support contracts. One would be administrative/secretarial—the other, technical. The technical "contract" could, in fact, be several contracts, each for a different sort of expertise, or it could be one contract awarded to a sufficiently diverse team.

The DBWG should be free to form subgroup structures as they are needed. These will most likely respond to different needs at different stages in the life of the DBWG activity. Initially, a subgroup structure should be formed that is oriented around the different kinds of issues presented in the Chapter 13. These issues can be grouped in ways that afford an opportunity for participants with similar interests and backgrounds to discuss a logical group of related issues. Thus, they can be described better, giving participants an improved understanding of their role in the entire DBWG effort. The objective of this initial organization would be to articulate and understand the reference model that would be used for the remainder of the group's activities. Later, a subgroup structure oriented around the products or around a set of orthogonal concerns would probably be more productive. One such structure might have a subgroup for each of Requirements, Available Technology (to meet the emerging requirements), and Approach (to formulate processes and considerations to be used in proceeding with the work).

One of the first activities of the DBWG should be to formulate a charter. This activity would focus and channel the thinking of the participants. Any subgroups should also formulate charters for their special objectives.

15.0 AVAILABLE TECHNOLOGY

No currently existing standard adequately addresses all of the DBMSIF concerns previously discussed here. However, a great deal of DBMSIF-related expertise exists in government, universities, and industry. The level of work being done by these various groups ranges from purely theoretical to attempts to produce products. These groups could provide potentially valuable input for developing the DBMSIF when the DBWG convenes in FY 92. A list of DBMSIF-related expertise in universities and industry will be compiled in FY 91.

16.0 TECIINICAL GROUPS

The "white paper" was forwarded to the following list of government and NGCR Operating System contacts for review at the end of 1990. All comments received have been incorporated in this final version of the document.

NAME	POC
SPAWAR 32432E	Comdr. David Hogen SPAWAR 32432 Washington, DC 20363-5100 (703) 602-9207 AV 332-9207
SPAWAR 3241E	Norma A. Stopyra SPAWAR 3241E Washington, DC 20363-5100 (703) 602-3966 AV 332-3966 stopyra@a.isi.edu
SPAWAR 3241	Philip J. Andrews SPAWAR 3241 Washington, DC 20363-5100 (703) 692-3966 AV 332-3966 pjandrews@a.isi.edu
SPAWAR 32431	Frank Deckelman SPAWAR 32431 Washington, DC 20363-5100 (703) 692-3966 AV 332-3966 deckelman@a.isi.edu
NARDAC—Navy Regional Data Automation Command	Robert A. Cooney Head, Data System Project Division NARDAC Code 4211, Bldg 143 Washington Navy Yard Washington, DC 20374-1435 (202) 433-2753 AV 288-2753 cooney@wnyose.nardac-dc.navy-mil

NUSC—Naval Underwater Systems Center	Tom Conrad NUSC Code 2221, Bldg 1171-3 Newport, RI 02841-5047 (401) 841-3354 AV 948-3354 tconrad@nusc-ada.arpa
JIAW—Joint Integrated Avionics	Ed Evers, General Dynamics Working Group Data Systems Division, 12101 Woodcrest Executive Drive P.O. Box 27366 St. Louis, MO 63141 (314) 851-8910
NSWC—Naval Surface Weapons Center	Daniel Green NSWC Dahlgren, VA 22448 (703) 663-4585 dtgreen@NSWC.navy.mil
NADC—Naval Air Development Center	Patricia Oberndorf NADC Code 7031 Warminster, PA 18974-5000 (215) 441-2737/Av 441-2737 tricia@nadc.nadc.navy.mil
Mitre Corporation	Anthony Carangelo, Jr. MS B325 Trusted Computer Systems The Mitre Corporation Burlington Rd. Bedford, MA 01730 (617) 271-3295 ac@security.mitre.org
NIST—National Institute of Standards	Gary Fisher NIST National Computer Systems Laboratory 225 Technology Bldg Rm B266 Gaithersburg, MD 20899 (301) 975-3275

Air Force—HQ AFSC/ENR	Capt. Peter M. Vaccaro HQ AFSC/ENR Andrews AFB Washington, DC 20334-5000 (301)981-6941
IDA—Institute of Defense Analysis	Dr. Karen D. Gordon IDA/CSED 1801 N. Beauregard St. Alexandria, VA 22311 gordon@ida.org
GTE—General Telephone	Andy Bibain Electronics GTE Irving, TX arb1@bunny.gte.com
General Dynamics	David Kellogg General Dynamics Fort Worth, TX kellogg@nadc.nadc.navy.mil 817-762-8017
Coast Guard	Comdr. Rex Buddenberg budden@manta.nosc.mil

17.0 REFERENCES

Ada 9X Report. 1990. "Realtime issues for Ada." August 1990.

American National Standards Institute (ANSI): Database Language SQL, X3.135-1986. (Also available as ISO Document ISO TC97/SC21/WG3 N117)

Boland, T. 1989. "Government Open Systems Interconnection Profile Users Guide (Draft Report)," National Institute of Standards and Technology, Gaithersburg, MD.

Butterbrodt, M. 1990. "FY 90 Final Report on Distributed Database for Distributed Command Control." NOSC Technical Document 2110, May 1990. Naval Ocean Systems Center, San Diego, CA 92152-5000.

Butterbrodt, M. C., and J. N. Baker. 1988a. "Analysis on the Feasibility of an NTDS Shadow Database." NOSC Technical Note 1558, November 1988. Naval Ocean Systems Center, San Diego, CA 92152-5000.*

Butterbrodt, M. C., and J. N. Baker. 1988b. "Analysis on the Validity of an NTDS Shadow Database." NOSC Technical Note 1559, November 1988. Naval Ocean Systems Center, San Diego, CA 92152-5000.*

Butterbrodt, M. C., and J. N. Baker. 1988c. "Naval Tactical Data System (NTDS) Shadow Database Project." NOSC Technical Document 1603, November 1988. Naval Ocean Systems Center, San Diego, CA 92152-5000.

Butterbrodt, M. C., and J. C. M. Green. 1990. "The Distributed Operating System Experiment: A Vital Link in the Development of a Realtime Relational Database Environment." CDRL Item produced January 1990, Naval Ocean Systems Center, San Diego, CA 92152-5000. (Available upon request)

Computer Corporation of America. 1982. "Global Query Optimization in MULTIBASE, A System for Heterogeneous Distributed Databases."

Davis, L. 1990. "The Promise of Network Databases," August 1990. *Datamation.*

DeWitt D. and H. Boral. 1984. "Methodology for Database System Performance Evaluation." University of Wisconsin Computer Sciences Technical Report.

GOSIP. 1988. Government Open Systems Interconnection Profile Users Guide.

Harmon, P., and D. King. 1985. "Expert Systems."

Hart, D. 1990. "X.500 Directory Product To Ship," *UNIX Today*, 25 June 1990.

ISO 9579-1. 1989. "Information Processing Systems—Open Systems Interconnection—Remote Database Access, Generic Remote Database Access Service and Protocol."

* NOSC Technical Notes (TNs) are working documents and do not represent an official policy statement of the Naval Ocean Systems Center. For further information, contact the author.

ISO 9579-2. 1989. "Information Processing Systems—Open Systems Interconnection—Remote Database Access, SQL Specialization."

ISO IS 9594-2 1987. "Information Processing Systems—Open Systems Interconnection—The Directory—Part 2: Information Framework."

ISO IS 9594-3. 1987. "Information Processing Systems—Open Systems Interconnection—The Directory—Part 3: Access and System Services Definition."

ISO IS 9594-4. 1987. "Information Processing Systems—Open Systems Interconnection—The Directory—Part 4: Procedures for Distributed Operation."

ISO IS 9594-5. 1987. "Information Processing Systems—Open Systems Interconnection—The Directory—Part 5: Access and System Protocols Specification."

ISO IS 9594-6. 1987. "Information Processing Systems—Open Systems Interconnection—The Directory—Part 6: Selected Attribute Types."

ISO IS 9594-7. 1987. "Information Processing Systems—Open Systems Interconnection—The Directory—Part 7: Selected Object Classes."

Jackson, K. 1990. "Countdown to GOSIP," *Communications Week*, 23 July 1990, pp. 1, 19-20.

Kamel, Nagdi. 1990. "Interoperability and Integration in Heterogeneous Database Environments."

Maslin, M., and B. Sundheim. 1990. "Natural-Language Interfaces to Navy Command and Control Databases, An Evaluation." NOSC Technical Document 1948, September 1990. Naval Ocean Systems Center, San Diego, CA 92152-5000.

McCarthy, J., et al. 1962. "LISP 1.5 Programmer's Manual," published by the MIT Press, August 1962, Cambridge 39, Massachusetts.

Mollet, A. A. 1990. "Distributed Query Processor Research." NOSC Technical Document 2053, produced by Computer Sciences Corporation, October 1990, for NOSC Code 413, San Diego, CA 92152-5000.

NCSC. 1990. Third draft of a Trusted Database Interpretation (TDI) for its Orange Book, officially the "Trusted Computer Security Evaluation Criteria," October 1990.

NGCR OSSWG Operational Concepts Document (Draft), Version 0.2, July 1990.

NGCR OSSWG Recommendation Report, May 1990.

ObjectStore. 1990. Object Design, November 1990, One New England Executive Park, Burlington, Massachussets 01803.

PROLOG and Databases, Implementations and New Directions. 1988. Published by Halstead Press of John Wiley & Sons

Ritchie, D., and B. Kernighan. 1978. "The 'C' Programming Language," Bell Telephone Laboratories.

SAFENET Standard B, Revision B, June 1990.

Saunders, R., and A. Weaver. 1990. "The Xpress Transfer Protocol (XTP)—A Tutorial," Computer Networks Laboratory, Department of Computer Science, University of Virginia.

Small, D. 1990. "Distributed Realtime Database System Navy Command/Control Systems—Afloat (NCCS-A) Project." NOSC Technical Document 1798, April 1990. Naval Ocean Systems Center, San Diego, CA 92152-5000.

UniForum. 1989. "Your Guide to POSIX."

Warnock, Fred. 1990. "Nuclear Survivability," Naval Air Development Center, Warminster, Pennsylvania.

APPENDIX A

ACRONYMS

APPENDIX A
ACRONYMS

Appendix A spells out the acronyms used in this document.

ACCAT	Advanced Command Control Architectural Testbed
ACDS	Advanced Combat Direction System
ACS	Afloat Correlation System
ACSE	Association Control Service Element
AEP	Application Environment Profiles
ANSI	American National Standards Institute
API	Application Programming Interface
CCA	Computer Corporation of America
CD ROM	Compact Disk Read Only Memory
CINCLANTFLT	Commander in Chief Atlantic Fleet
CINCPACFLT	Commander in Chief Pacific Fleet
CMS-2	Compiler Monitor System, Version 2
COBOL	Common Business Oriented Programming Language
DAC/MAC	Discretionary/Mandatory Access Control
DARPA	Defense Advanced Research Projects Agency
DBMS	Database Management System
DBMSIF	Database Management System Interface Standard
DBWG	Database Management System Interface Working Group
DEC	Digital Equipment Corporation
DECNET	Digital Equipment Corporation Network
DIA	Defense Intelligence Agency
DIB	Directory Information Base
DQP	Distributed Query Processor
EMSP	Enhanced Modular Signal Processor, AN/UYS-2
FAP	Standard Formats and Protocols
FCCBMP	Fleet Command Control Battle Management Program
FDDI	Fiber Distributed Data Interface

FIPS	Federal Information Processing Standard
FORSTAT	Force Status
FORTRAN	Formula Translation Programming Language
FTAM	File Transfer, Access, and Management
FY	Fiscal Year
GOSIP	U.S. Government Open Systems Interconnection [OSI] Profile
HDLC LAPB	High Level Data Link Control (HDLC) Link Access Procedure B (LAP B)
IBM	International Business Machines
IS	International Standard
ISO	International Standards Organization
ISODE	ISO Development Environment
JIAWG	Joint Integrated Avionics Working Group
LAN	Local Area Network
LHA	Amphibious Assault Ship, General Purpose
LHD	Landing Helicopter Dock Amphibious Ship
LISP	List Processing Language
LLC	Logical Link Control
MIIDS	Military Intelligence Integrated Database System
MIS	Management Information System
MOVEREP	Movement Reports
NASA	National Aeronautics and Space Administration
NASEE	Naval Air Systems Command Software Engineering Environment Toolset
NCSC	National Computer Security Center
NGCR	Next Generation Computer Resources
NIST	National Institute of Standards and Technology
NLI	Natural Language Interface
NLU	Natural Language Understanding
NOSC	Naval Ocean Systems Center
NSWC	Naval Surface Weapons Center

NTCS-A	Naval Tactical Command Systems-Afloat
NTDS	Naval Tactical Data System
NWSS	Navy WWMCCS Standard System
NWTDB	Naval Warfare Tactical Database
OO DB	Object Oriented Data Base
OS	Operating System
OSI	Open Systems Interconnection
OSSWG	Operating Systems Standards Working Group
PC	Personal Computer
POSIX	Portable Operating System Interface for Computer Environments
RDA	Remote Database Access
SAFENET	Survivable Adaptable Fiber-Optic Embedded Network
SNA	Systems Network Architecture
SQL	Structured Query Language
TADSTAND	Tactical Digital Standards
TCB	Trusted Computing Base
TCP/IP	Transmission Control Protocol/Internet Protocol
TDBMS	Trusted Database Management System
TDI	Trusted Database Interpretation
TDS	Tactical Data System
TP1	Transaction Processing Standard 1
UniForum	International Association of UNIX Systems Users
VT	Virtual Terminal
WWMCCS	World Wide Military Command Control System
XTP	Xpress Transfer Protocol
X.400	Message Handling System (MHS or X.400)

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1991	Final: November 1989—May 1991

4. TITLE AND SUBTITLE

WHITE PAPER ON THE DATABASE MANAGEMENT SYSTEM INTERFACE STANDARD FOR NAVY NEXT GENERATION COMPUTING RESOURCES (NGCR)

5. FUNDING NUMBERS

Proj. No. 41–CC30 01
LWU: DN587574
PE: OMN

6. AUTHOR(S)

D. L. Small, M. C. Butterbrodt, R. M. Bergman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Ocean Systems Center
San Diego, CA 92152–5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NOSC TD 2116

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space and Naval Warfare Systems Command
Washington, DC 20363–5100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A white paper (technical document) presents the advanced analysis, investigation, and planning regarding the issues facing the Database Management System Interface Working Group (DBWG) involved with the Navy's Next Generation Computing Resources (NGCR) program. It also provides a starting point for discussions when the DBWG is initiated in FY 92. At the time of its writing, this document represents a "snapshot" of the Navy's use of a database management system (DBMS) technology for command, control, and combat systems.

14. SUBJECT TERMS

interface standards
computing resources
Navy C^3I systems
DBMS interface requirements

15. NUMBER OF PAGES

65

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

UNCLASSIFIED

21a. NAME OF RESPONSIBLE INDIVIDUAL	21b. TELEPHONE *(Include Area Code)*	21c. OFFICE SYMBOL
D.L. Small	(619) 553-4130	Code 413

NSN 7540-01-280-5500

Standard form 298

UNCLASSIFIED

INITIAL DISTRIBUTION

Code 0012	Patent Counsel	(1)
Code 0144	R. November	(1)
Code 40	R. C. Kolb	(1)
Code 405	P. Garcia	(1)
Code 41	A. Justice	(1)
Code 41	G. Myers	(1)
Code 41	M. Crowley	(1)
Code 411	L. Sutton	(1)
Code 412	L. Core	(1)
Code 412	R. Bergman	(25)
Code 412	D. Fisher	(1)
Code 413	M. Butterbrodt	(1)
Code 413	A. Thanner	(1)
Code 413	M. Thumpold	(1)
Code 413	L. Anderson	(1)
Code 413	M. Gadbois	(1)
Code 413	D. Brouhard	(1)
Code 413	R. Johnston	(1)
Code 413	L. Dunham	(1)
Code 413	J. Drummond	(1)
Code 413	D. Small	(25)
Code 423	M. Ceruti	(1)
Code 433	B. Sonnenfeld	(1)
Code 433	R. Delgado	(1)
Code 441	L. Wong	(1)
Code 444	B. Sundheim	(1)
Code 444	D. Eddington	(1)
Code 444	R. Dillard	(1)
Code 444	T. Wadsworth	(1)
Code 621	B. Lubach	(1)
Code 655	B. Edwards	(1)
Code 771	B. Pollett	(1)
Code 8601	Lt D. Welke	(1)
Code 925	B. Edwards	(1)

Defense Technical Information Center
Alexandria, VA 22304-6145 (4)

NOSC Liaison Office
Washington, DC 20363-5100

Center for Naval Analyses
Alexandria, VA 22302-0268

Navy Acquisition, Research & Development
Information Center (NARDIC)
Alexandria, VA 22333

National Institute of Standards & Technology
Gaithersburg, MD 20899

Defense Communications Agency (DCA)
Arlington, VA 22203

Space & Naval Warfare Systems Command
Washington, DC 20363-5100 (6)

Naval Air Development Center
Warminster, PA 18974-5000

Naval Surface Warfare Center
Dahlgren, VA 22448

Naval Underwater Systems Center
Newport, RI 02841-5047

Navy Regional Data Automation Command
Washington Navy Yard
Washington, DC 20374-1435

Office of Naval Research
Arlington, VA 22217-5000

Naval Sea Systems Command
Washington, DC 20362-5101

U.S. Coast Guard
Washington, DC 20593

U.S. Coast Guard
Wildwood, NJ 08260

Rome Air Development Center
Griffiss Air Force Base, NY 13441

HQ AFSC/ENR
Andrews AFB
Washingotn, DC 20334-5000

Institute of Defense Analysis
Alexandria, VA 22311

Joint Integrated Avionics Working Group
c/o General Dynamics
St Louis, MO 63141

BBN Communications Corporation
Columbia, MD 21046

Computer Sciences Corporation
San Diego, CA 92110-5164 (2)

Evolutionary Technologies, Inc.
Austin, TX 78720-3372

GE Aerospace
Philadelphia, PA 19101

General Dynamics
Fort Worth, TX 76101

General Telephone Electronics (GTE)
Irving, TX 75207

Hughes Corporation
Fullerton, CA 92634-3310

IBM Corporation
Manassas, VA 22110

The MITRE Corporation
Bedford, MA 01730

SYBASE
La Jolla, CA 92037

Westinghouse Electric Corporation
Baltimore, MD 21203