AD-A262 856

REPORT DOC

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1992

3. REPORT TYPE AND DATES COVERED
professional paper

4. TITLE AND SUBTITLE
OPERATIONS SUPPORT SYSTEM (OSS) INTEGRATED DATABASE (IDB) DESIGN AND DEVELOPMENT: SOFTWARE REUSE LESSONS LEARNED

5. FUNDING NUMBERS
In-house

6. AUTHOR(S)
M. G. Ceruti and S. D. Rotter

DTIC
S ELECTE D
APR 1 2 1993
C

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division (NRaD)
San Diego, CA 92152–5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Naval Command, Control and Ocean Surveillance Center (NCCOSC)
RDT&E Division (NRaD)
San Diego, CA 92152–5000

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

93-07479

13. ABSTRACT *(Maximum 200 words)*

Lessons learned are presented concerning the reuse of information management systems components in the design, development and support of the Operations Support System (OSS) Integrated Database (IDB). These components were developed originally under a wide variety of C^4I programs of Navy, Air Force and Joint Command Centers. Reuse of components was possible because known standards such as the C programming language, Structured Query Language (SQL) and the Naval Warfare Tactical Database (NWTDB) were used during development. Reused components included data, data structure, catalogue scheme for defining data structure, database applications, and software tools used to validate data integrity. It was found that strict adherence to standards was the key to successful software and data reuse. Other lessons learned included that early design of software modules should be done with a view toward future reuse. The portability and generality of American Standard Code for Information Interchange (ASCII) flat files that are not vendor specific enhance information reuse, whereas a relational database management system is more advantageous when performing operations associated with the relationships between tables. Because of the interoperability and efficiency obtained from software reuse, the OSS program derived substantial benefits from this effort.

93 4 09 021

Published in Proceedings of the AFCEA Database Colloqium '92, August 1992.

14. SUBJECT TERMS
Operations Support System (OSS)
Integrated Database (IDB)
Joint Operations Tactical System (JOTS)

15. NUMBER OF PAGES

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
Dr. M. G. Ceruti	(619) 553-4068	Code 423

NSN 7540-01-280-5500

Standard form 298 (BACK)

UNCLASSIFIED

Operations Support System (OSS) Integrated Database (IDB) Design and Development: Software Reuse Lessons Learned

Dr. Marion G. Ceruti and Sharon D. Rotter
Command and Control Department,
Naval Command, Control and Ocean Surveillance Center,
Research, Development, Test, and Evaluation Division,
San Diego, CA 92152-5000

&

Kristofer Timmerman and Jennifer Ross
Science Applications International Corporation,
Comsystems Division
10770 Wateridge Circle, San Diego, CA, 92121

ABSTRACT

Lessons learned are presented concerning the reuse of information management systems components in the design, development and support of the Operations Support System (OSS) Integrated Database (IDB). These components were developed originally under a wide variety of C^4I programs for Navy, Air Force and Joint Command Centers. Reuse of components was possible because known standards such as the C programming language, Structured Query Language (SQL) and the Naval Warfare Tactical Database (NWTDB) were used during development. Reused components included data, data structure, catalogue scheme for defining data structure, database applications, and software tools used to validate data integrity. It was found that strict adherence to standards was the key to successful software and data reuse. Other lessons learned included that early design of software modules should be done with a view toward future reuse. The portability and generality of American Standard Code for Information Interchange (ASCII) flat files that are not vendor specific enhance information reuse, whereas a relational database management system is more advantageous when performing operations associated with the relationships between tables. Because of the interoperability and efficiency obtained from software reuse, the OSS program derived substantial benefits from this effort.

Proceedings of the AFCEA Database Colloquium '92 on 25-27 August 1992

OSS hardware and software have been installed in military command centers throughout the world, including the Chief of Naval Operations (CNO) Command Center, Washington, DC; the Commander in Chief, Pacific Fleet (CINCPACFLT); the United States Commander in Chief, Pacific Forces (USCINCPAC); the Commander in Chief, Pacific Air Forces (CINCPACAF); the Commander in Chief, Atlantic Fleet (CINCLANTFLT); the United States Commander in Chief, Atlantic (USCINCLANT); the Strategic Air Command, Atlantic (SACLANT); the Commander in Chief, United States Naval Forces, Europe, (CINCUSNAVEUR); the Joint Operations Command Center, Naples, Italy; and the Commander, Iceland Defense Force, Keflavik, Iceland.

The OSS IDB supports a variety of command center functions, including display, message traffic, platform position tracking, planning, and capabilities analysis. It consists of several databases, including data supporting OSS-unique requirements. These databases are described below in Section 3.0, on the OSS Integrated Database.

Software reuse methodology as implemented in the Software Reuse Library is also an integral part of the OSS program. Software reuse practices have enabled OSS to develop more efficiently by avoiding costly duplication of effort. Standards essential for software reuse are important in ensuring system interoperability. Thus the advantages of software reuse are not limited to cost considerations.

2.0 THE DATABASE-CENTRIC MODEL

The quality of databases accessed by C^4I systems is the most important factor in determining the validity and reliability of the C^4I system functions. This concept is captured in the Database Centric Model (Schill, 1989), which has characterized the philosophy of development of the OSS Integrated Database (IDB) as it relates to command center operations. The Database-Centric model of command and control includes all of the functions of a command center and is based on the concept that the database is of central importance in a command center.

Outgoing messages, such as those from the Joint Operations Tactical System (JOTS), which is an OSS component, are sent using database information, and inbound messages are parsed into the database. Data can be displayed in several formats, including work station monitors, large-group displays, and hard-copy printouts.

3.0 OSS INTEGRATED DATABASE

Design

The structure of the OSS Integrated Database (IDB) was patterned after databases developed for several other Department-of-Defense programs. A substantial part of the structure of the OSS IDB was designed in accordance with the Naval Warfare Tactical Database (NWTDB), the authoritative information source for all Naval warfare systems. NWTDB is maintained and distributed according to the information architecture specified in the NWTDB Standards and Structure Encyclopedia. The Naval *Intelligence Database (NID), managed and maintained* by the Naval Maritime Intelligence Center, constitutes Volume I of the NWTDB Standards and Structure Encyclopedia. The NID is the Navy's authoritative database of selected technical characteristics and performance data for platforms, weapons, and sensors (Black, 1989; NAVINTCOM, 1990).

Databases from several other C^4I programs were major influences on the design of the OSS IDB. These programs included the Fleet Command Center Battle Management Program (FCCBMP), the Maritime Defense Zone (MDZ), the CINCPACAF Integrated Decision Support System (CIDSS), and the OSIS Baseline Upgrade (OBU).

Data Sets

The OSS IDB consists of approximately 560 tables, and several categories of data sets. Each table is assigned to a category in the OSS Integrated Data Dictionary. Because of the modularity of these clearly defined categories, a developer can ascertain the tables needed for porting a particular segment of software code to a new system.

In addition to the NID, data sets in the OSS IDB are derived from several other sources of static and dynamic data. One of the principal objectives of OSS is to give users access to positional, readiness, and casualty data. Positional data on ships from JOTS enter the database through an automatic data link at the command center sites. Status of Readiness and Training Systems (SORTS) readiness data, movement reports, and casualty reports are also input via messages.

NWSS messages concerning employment schedules are generated at command centers, and these data fill tables specially designed to support

The CRSS library contains over 150 megabytes of reusable software and the number of available components is growing. Software developers can use existing components to speed software development and to reduce costs by eliminating redundant effort by enforcing standards to promote a common look and feel. The CRSS library has access to several database development tools that were used for OSS and other C^4I programs (Ceruti and Auclair, 1991).

Advantages of Software Reuse

Reuse of software components througho it the life-cycle can improve software development productivity and increase the quality of software products. When a software engineer selects an existing component from a repository and incorporates that component with little or no modification into a new application, the software developer is free to perform more complex tasks, thus improving overall productivity. By allowing the developer to write fewer lines of text or code, the reuse of components amplifies the developer's capabilities (Biggerstaff and Richter, 1987).

Reuse is not limited to source code; cost savings also result from the reuse of other entities such as design representations and test routines. The economic advantages of software reuse are evident during the entire software life cycle, including testing. Costs are reduced from initial development to long-term maintenance. Moreover, if a component is tested and certified to some degree prior to being placed in a repository and is reused and retested, it is reasonable to assume that the quality of the component would increase through repeated testing and reuse.

Components are more interoperable when standards facilitate their reused. When the reused component is for example, a user interface to the database, this advantage is extended to include human factors as well. Users of systems that have a common look and feel find that less time is required to grasp the mechanics of the interface, thus improving efficiency.

5.0 SOFTWARE REUSE IN THE OSS IDB

Databases and related software for several C^4I programs were developed in the Information Management Engineering (IME) Laboratory, located at NRaD (Ceruti and Auclair, 1991). Because the database requirements in OSS were similar to those of other C^4I systems, some information management system components from other projects were reused in the OSS IDB design and development.

Reuse of Database Software Tools in the UNIX Operating System.

Many of the reused components were database-related software tools such as command files originally written in the Virtual Address eXtension/Virtual Memory System (VAX/VMS) Digital Command Language (DCL) and Standard Query Language (SQL). These software tools, developed originally to execute on computers manufactured by the Digital Equipment Corporation (DEC), were reused during the development of the OSS IDB. The majority of these tools were created by database developers to aid in controlling data, tracking progress, detecting errors, and ensuring data integrity. Because UNIX was adopted as the standard operating system for OSS, information management components were transferred to a UNIX environment to support further development of the OSS IDB.

Although not an industry standard, Oracle was the RDBMS used for the development of the OSS IDB. The databases of related C^4I programs such as FCCBMP, CIDSS, MDZ, OBU, and the Operations Support Group Prototype (OSGP), also were developed under VMS and Oracle (Ceruti and Auclair, 1991). Having an RDBMS common to both VMS and UNIX greatly facilitated the reuse of existing software code and data from these programs.

The reused software tools include the following:

Table and Index Creation. Creates or removes tables as catalogued in data dictionary schema.

Row Count Report. Generates a report of the number of records in every table in database.

Percent Fill. Generates a report of the percentage of non-null values in every attribute of every table in the database.

Data and data structure from the NID also were reused in the OSS IDB with a minimum of modification. Names of newly developed relations and attributes were selected according to NWTDB standard naming conventions. Relation names were chosen to appear similar or identical to those of existing NID relations. For example, new attribute names in an IDB relation were chosen to be the same as those found in the NID if the attribute definitions and data types were the same as those found in the NID. Seven metadata elements containing source information are found in all NID relations (NAVINTCOM, 1990). These attributes, AGENCY_SOURCE_CODE, AGENCY_COGNIZANT_CODE, DATE_ROW_LOAD, DATE_ROW_CHANGE, DATE_ROW_SOURCE, SEC_CLASS_CODE and CONF_LVL_CODE, were added to most relations in the OSS IDB for audit and source tracking. Because the standard NID relation and attribute naming conventions were used for new and existing database entities in the OSS IDB, no changes in structure or fill were necessary to integrate the NID data into the OSS IDB.

Developing Reusable Code

The following is a description of some software tools that were developed in the UNIX environment with the idea of reuse in mind:

Automatic Installation. Loads Oracle RDBMS onto a machine and imports the OSS IDB with a minimum of user attention.

Instance Creation. Creates an instance of Oracle with a minimum of user attention.

Data Dictionary/System Comparison. Compares metadata in data dictionary tables with actual Oracle table structure to ensure exact correspondence.

Table Size Calculation. Determines the appropriate table sizes for all relations in the OSS IDB and stores the results on a disk for future input into table creation software.

advantages of flat files with those of RDBMS-based tables. If such a standard export format does not evolve, the next best thing, in cases where a commercial RDBMS is used, is to restrict the implementation to ANSI SQL in applications that need to be ported from one RDBMS to the next. This would go a long way toward eliminating obstacles to interoperability and portability such as vendor-specific enhancements to SQL not found in other RDBMSs.

Conclusion

As with the software development process, a model leading to an ingrained level of reuse maturity must be established within the organization. Reuse must be raised from a chaotic, ad-hoc level to one in which reuse is indoctrinated as "the way we do business". Project goals must include a strategic plan to incorporate reuse early and throughout the project life-cycle and must result in software products generic enough for future reuse. Software components should be treated as key capital assets leading to a competitive advantage and a leadership position in the marketplace. Reuse methodology must be fully integrated with the software development process and in the software development environment.

AUTHOR BIOGRAPHIES

Dr. Marion G. Ceruti
NCCOSC RDT&E Division, San Diego, CA

Dr. Marion G. Ceruti is a scientist in the Advanced C^3I Systems Branch, Code 423, at the Naval Command, Control and Ocean Surveillance Center, Research, Development, Test, and Evaluation Division. She received her Ph.D. in Physical Chemistry, with emphasis on data acquisition systems, from the University of California at Los Angeles (UCLA) in 1979. While at UCLA, Dr. Ceruti was awarded a research fellowship from the International Business Machine Corporation. Her present professional activities include database development task leader and coordinator for C^4I decision support systems, including the Operations Support System. Dr. Ceruti has served on the program committee and as Government Point of Contact for all annual Database Colloquia since 1987. An active member of AFCEA and several other scientific and professional organizations, Dr. Ceruti is the author of publications on various topics in science and engineering, including information management.

Sharon D. Rotter
NCCOSC RDT&E Division, San Diego, CA

Sharon D. Rotter of the Software Technology Branch, Code 411, received the NRaD Exemplary Achievement Award in 1992 for her significant contributions to the development of the C^3I Reusable Software System (CRSS), a reuse library containing command and control components, which is being developed under the charter of the Operations Support System. Ms. Rotter is also the principal investigator of a research project sponsored by the Office of Naval Technology investigating domain analysis and engineering methodologies applied to Navy C^3I systems. She was the program manager for the development of a rapid prototype for U. S. Message Text Format (USMTF) origination and validation developed in Ada and targeted for the World Wide Military Command and Control System (WWMCCS) community. Ms. Rotter is a cum laude graduate of San Diego State University with a B.S. degree in Information Systems, with emphasis on Systems Analysis, and is the President of the local chapter of the Special Interest Group for Ada (SIGAda). She is also a member of AFCEA, the Association of Computing Machinery (ACM) and the Special Interest Group for Software Engineering (SIGSOFT).