

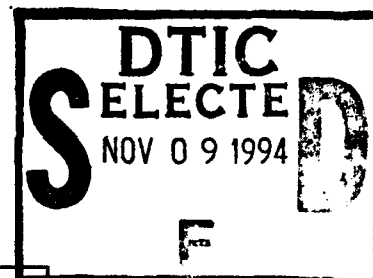
AD-A286 021

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS



EFFECTIVE MANAGEMENT OF CLASSIFIED DOCUMENTS USING THE LIBRARY DOCUMENT SYSTEM

by

Kenneth F. Elkern, Jr.

September, 1994

Thesis Advisor:

Timothy J. Shimeall

Approved for public release; distribution is unlimited.

94-34603



DTIC QUALITY INSPECTED 1

94 11 7 021

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE Sep 1994		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE Effective Management of Classified Documents Using the Library Document System			5. FUNDING NUMBERS	
6. AUTHOR(S) Elkern, Kenneth, F. Jr.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Cheif of Naval Education and Training, Pensacola, FL			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) Previous automated classified document systems developed commercially or in-house to serve classified libraries with 50,000 documents or less, have been limited by excessive cost or insufficient functionality. The problem addressed by this research was to improve the automated systems available to classified libraries with 50,000 documents or less, by upgrading the Library Document System (LDS) to meet the tracking and document search needs of librarians. The approach taken was to first conduct a survey of 25 classified libraries to assess their document tracking procedures and requirements. Next, a thorough examination of the commercial and in-house automated classified document systems was performed to determine the state of solutions available. Finally, a strategy for modifying LDS was outlined to incorporate the tracking and document search features desired, using modern relational database constructs, structured programming techniques, and user-friendly interface design. As a result of this work, LDS was upgraded to fulfill the needs of classified librarians with 50,000 documents or less. In particular the schemata of the system were extended, a sophisticated search facility was implemented, and a mouse-oriented user-friendly interface was provided.				
14. SUBJECT TERMS Computer software, Library Automation			15. NUMBER OF PAGES 90	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Approved for public release; distribution is unlimited.

Effective Management of Classified
Documents Using the Library Document System

by

Kenneth F. Elkern, Jr.
Lieutenant , United States Navy
B.A., Illinois State University, 1987

Submitted in partial fulfillment
of the requirements for the degree of

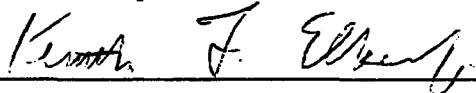
MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

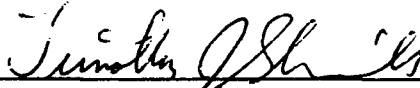
September 1994

Author:



Kenneth F. Elkern Jr.

Approved by:



Timothy J. Shimeall, Thesis Advisor



David Gaitros, Second Reader



Ted Lewis, Chairman
Department of Computer Science

ABSTRACT

Previous automated classified document systems developed commercially or in-house to serve classified libraries with 50,000 documents or less, have been limited by excessive cost or insufficient functionality. The problem addressed by this research was to improve the automated systems available to classified libraries with 50,000 documents or less, by upgrading the Library Document System (LDS) to meet the tracking and document search needs of librarians.

The approach taken was to first conduct a survey of 25 classified libraries to assess their document tracking procedures and requirements. Next, a thorough examination of the commercial and in-house automated classified document systems was performed to determine the state of solutions available. Finally, a strategy for modifying LDS was outlined to incorporate the tracking and document search features desired, using modern relational database constructs, structured programming techniques, and user-friendly interface design.

As a result of this work, LDS was upgraded to fulfill the needs of classified librarians with 50,000 documents or less. In particular the schemata of the system were extended, a sophisticated search facility was implemented, and a mouse-oriented user-friendly interface was provided.

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

TABLE OF CONTENTS

I. INTRODUCTION	1
A. BACKGROUND	1
B. PROBLEM STATEMENT	3
1. Update LDS to use modern computer technology	3
2. Generalize LDS from GENSER to more global environment	4
3. Research Issues	4
C. APPROACH	5
D. OVERVIEW OF THESIS	8
II. RESEARCH PHASE	9
A. SURVEY JUSTIFICATION	9
B. COMMANDS SELECTED FOR THE SURVEY	11
C. SCOPE OF SURVEY	13
D. SURVEY RESULTS	13
E. ALTERNATIVE SYSTEMS	14
F. UPGRADE LDS	18
G. UPGRADE FUNCTIONS AND STRUCTURE	19
III. LDS SCHEMA IN DETAIL	21
A. DATABASE MANAGEMENT TECHNIQUES	21
B. ORIGINAL LDS SCHEMA	21
C. CDCS SCHEMA	23
D. NEW LDS SCHEMA	25
IV. SEARCH	30
A. JUSTIFICATION FOR SEARCH	30
B. PREVIOUS SEARCH	31
C. NEW SEARCH OBJECTIVES	33
D. SEARCH ALGORITHM	35
E. NEW SEARCH FEATURES	41
V. LDS EXAMPLE	45
A. BREADTH OF EXAMPLE	45
B. SEARCH EXAMPLE	45

C. CLASSIFICATIONS	50
D. CUSTODIANS	53
E. DOCUMENTS	56
F. ON-LINE HELP	61
VI. CONCLUSION	63
A. RESEARCH SUMMARY	63
B. APPLICATION	64
C. FUTURE RESEARCH	65
APPENDIX A: LDS SURVEY	67
APPENDIX B: LDS DATABASE SCHEMA	74
LIST OF REFERENCES	81
INITIAL DISTRIBUTION LIST	82

ACKNOWLEDGMENTS

This project is not the work of one individual alone. There are several people who contributed to this work both directly and indirectly. Although it would be impossible to acknowledge all of them, I would like to take this opportunity to thank the major players: Dr. Timothy J. Shimeall, who provided considerable support and encouragement throughout this work; LtCol Gaitros who assisted in making this thesis more understandable; Jim Bouzios who provided both computer hardware and software to make the implementation possible; Betty Snellenberg who provided additional computer hardware which expedited LDS development. Additionally, I would like to thank Mike Williams and Arnold Hankins for assisting in the survey process.

Finally, I would like to thank my wife, for her loving support and understanding throughout this work. Her assistance in proofreading was invaluable.

Without the help and moral support of the above people and countless others, this work would not have been possible.

I. INTRODUCTION

A. BACKGROUND

In 1989, as microcomputers were becoming more prevalent in the Navy, many commands discerned the potential benefits and opportunities for developing in-house systems to automate and streamline their operations. The Navy and Marine Corps Intelligence Training Center (NMITC) in Virginia Beach was one such command that sought to integrate the technological changes. Its classified library, with thousands of documents that were difficult and tedious to manage by hand, was a perfect candidate for computer automation. Their idea was simple: develop a classified document control system that would maintain the entire collection of documents while providing multi-user access through a Local Area Network (LAN). By creating a comprehensive automated database management system in-house, not only would the system be tailor made to NMITC's specifications, but it would save the Navy approximately \$500,000 in outside development costs. At that time, commercial library systems were primarily being developed for the unclassified public libraries and their costs were still prohibitive for small libraries such as NMITC's.

The Library Document System (LDS) was developed and tested over a six month period and became fully operational in November 1989. Because of its success and reliability, LDS has been distributed to over 30 commands in the Navy and two commands

in the Air Force. Since LDS was designed to operate in a GENSER¹ environment, its functionality is somewhat restricted when implemented in classification levels beyond GENSER.

In 1994, there is still a need for a low-cost, specialized system like LDS. Performing a critical role in the development and implementation of LDS, I have become acutely aware of the capabilities of automated classified document systems. As a Naval Intelligence Officer, I have observed a variety of these systems over the last five years on ships and at shore facilities. Since there is no Navy standardized system, library administrators must develop a method for managing their documents. This method may or may not be beneficial to the patrons the libraries are there to serve. Although, a simple tracking system may keep the administrator up-to-date on the whereabouts of any given document, if that system does not provide a query capability for patrons, it is of little use. Given that the amount of information that is generated and disseminated in this information age, devising a system that can assist patrons in eliminating excess or unneeded information is paramount. One solution to this situation is to formally define the needs of both the administrators and patrons and provide them with a system like LDS. However, with the technological advances, since 1989, coupled with the user's knowledge and exposure to intuitive user-interfaces, LDS must be upgraded with more sophisticated features and a more user friendly interface.

¹ Generic classification which includes unclassified, confidential and secret material.

B. PROBLEM STATEMENT

1. Update LDS to use modern computer technology

In the five years since LDS inception, technology has changed dramatically. LDS was designed to operate on a Zenith 248 (Intel 80286-8Mhz, 640K) running MS-DOS™ and LANTASTIC™ multi-user software. It was written in dBASE III PLUS™ and the CA-CLIPPER™ (Summer '87) procedural language. The computers and software systems available today are indisputably more advanced, performing hundreds of multi-processing tasks while operating in a multi-user environment. Although the previous versions of LDS remain upwardly compatible, no provisions were made for it to take advantage of the capabilities and resources provided by today's systems.

There are several areas that will be addressed when upgrading LDS to today's technology. The implementation of mouse support and more user friendly interface methods is essential for current and new users who are acclimated to using a mouse. The new version will need to take advantage of modern operating systems, LANs, and be programmed in an object-based structured programming language. LDS must have the capability to coexist with other applications sharing RAM and virtual memory resources without conflict. LDS should also account for hardware modifications such as: the multiple communication ports used for retrieving barcode data from portable barcode readers, interface with new barcode scanners, and the ability to address network and laser printers. Finally, if LDS is to continue to be successful, it must compete with larger, more expensive systems by accommodating user's demands. Such demands are document

abstracts for patrons to view on-screen before signing out documents and a Boolean search capability.

2. Generalize LDS from GENSER to more global environment

As discussed previously, LDS was designed and implemented to NMITC's GENSER specifications, its popularity with other commands was not anticipated. Therefore, integrating LDS at other commands was difficult when classification levels varied. Even though NATO classifications were made available with LDS in early 1990, users still wanted more flexibility, particularly the commands that work with higher classification levels or have their own set of unique classifications. To support a more flexible classification system, the introduction of custodians, audit trails and password protection mechanisms would be essential. If LDS is to appeal to a variety of commands and organizations DoD wide, then its classification's facility must be flexible and easy to use.

3. Research Issues

Several concerns were addressed and became the foundation for upgrading LDS. The first matter was to locate a Navy standard computerized system that facilitates the efficient tracking of classified documents. If such a system existed, it could function as a model for improving LDS. However, since there is no standard system in the Navy, investigating the systems command's currently use to manage their classified publications, would be the next logical step.

After locating several systems, it was necessary to explore the methods patrons use to access the library's classified documents. Unfortunately for patrons, many automated

tracking systems do little more than track document activity. Support modules that allow patrons to request documents on a specific subject are rarely incorporated into small systems.

Since classified libraries are not unique to the Navy, it was beneficial to consider the automated tracking systems used by other services. Although a complete evaluation of the automated systems used by the other services was beyond the scope of this thesis, those that participated provided some interesting results.

There are numerous commercial document tracking systems available that offer advanced features for a price. While investigating the commercial systems available, two issues were raised: could LDS be enhanced with the most significant features from commercial systems, and to what extent does LDS need to be improved to create the greatest user satisfaction and extend its usefulness into the future.

After resolving the above concerns, the movement toward developing a standardized system for the Navy was established.

C. APPROACH

In an attempt to determine the current state of automated classified document systems in the Navy, a survey was drafted and distributed to 25 commands (see Appendix A for actual survey and results). The results of the survey were conclusive: there is still a significant need for automated classified document systems and a replacement system needs to be found for an aging, unreliable program called the Classified Document Control

System (CDCS). There were a few commands that were either in the process of developing their own system or already had one in place. These commands possess the in-house expertise capable of developing a quality product. However, the majority of commands surveyed do not have the expertise nor the time required to develop a reliable system from the ground up and were notably dissatisfied with CDCS.

Following the survey, the latest version of CDCS (2.0) was acquired for a thorough analysis. The goal was two-fold; determine CDCS's compatibilities and/or incompatibilities with LDS and to gain an understanding of the users' discontentment with the program.

The next step was to contact commercial vendors of automated classified document systems and determine their cost-to-feature benefits over upgrading LDS. After contacting several vendors, it became clear that the costs outweigh the benefits for small libraries. The most reasonable system sold for \$45K for a stand-alone version, and upgrading to a network version cost \$45K per node. Therefore, due to budgetary constraints, the majority of these small libraries (50,000 documents or less) can not afford to purchase commercial systems. Library administrators are left with three options; use a logbook and hand write all transactions, use CDCS, or develop a system from scratch.

This situation is compounded when considering the effects on the patrons of these libraries. Unless the patron knows the exact document he/she is looking for, searching by topic may require thumbing through many documents. A more productive use of the patron's time would be the availability of an automated system that incorporates a flexible

search feature. Ideally the search feature would allow patrons and/or system administrators to locate documents on a particular subject quickly. Once the patron identifies the documents he/she wants, the automated system can assist the librarian in determining their current location.

Once the need to upgrade LDS was established, developing a systematic strategy took precedence. Several areas were outlined and documented as the most critical elements in upgrading LDS. A comprehensive analysis of CDCS revealed several features that were missing in LDS and must be included in the upgrade. The LDS '89 procedural code was evaluated and a method determined for rejuvenating it with structured programming constructs. LDS's new features: Boolean search capability, audit system, password protection, and the custodian and classification tables were profiled. Original portions of LDS such as the inventory and system reports were redesigned for efficiency and flexibility. Lastly, both survey results and current LDS user's comments were considered and factored into the upgrade process.

Three software packages were selected to accomplish the LDS upgrade: CA-Clipper 5.2 for its object-based structured coding framework, AAmouse for its mouse support routines, and Data Junction for its ability to convert from one database format to another literally streamlining the upgrade process for current CDCS or other database system's users.

D. OVERVIEW OF THESIS

Chapter II provides more background information on this thesis. In-depth information is presented on the survey results, alternative system solutions, and the previous versions of LDS. Additionally the functions and program structures designed to track classified documents explained. Chapter III discusses the LDS database schema and how its redesign efficiently uses disk storage space. Chapter IV describes the Boolean search feature in detail. Chapter V presents an extended example of how LDS functions from a user's perspective. Chapter VI summarizes the upgrade of LDS and future work.

II. RESEARCH PHASE

A. SURVEY JUSTIFICATION

The first step in researching the LDS upgrade took the form of a survey. Distributing a survey into the field satisfied three major objectives: to substantiate the need for an in-house developed automated classified documents systems (ACDSs), to inform current Classified Document Control System (CDCS) users that LDS, a supported DoN product, is being upgraded, and to acquire the pertinent features essential in systematically operating a classified library more efficiently. The results would provide a measure of the current state of ACDSs available in the Navy [Ref. 6]. The survey would also sustain or weaken information received early on in the research phase about suspicions concerning the functionality, reliability and technical support of the previously Navy sanctioned CDCS.

Notifying the CDCS users that work was in progress to upgrade LDS, a system that has been in operation for five years, was significant, since CDCS will no longer be supported. This being the case many commands have started writing their own software with or without qualified in-house personnel. The end result may be a ACDS that has flaws or is poorly designed. Other commands that do not have the in-house talent are left with either maintaining CDCS, going back to a hand system or allocating the funds to purchase a commercially developed system. With budget constraints in all services, the

latter option is not a practical one. Therefore, the information provided by the survey would immediately benefit those commands that were either planning or in the process of developing their own in-house system. These commands would simply need to wait for the release of the upgraded LDS.

The survey would provide a means for the library administrators to share their opinions about which elements, they feel, create a reliable ACDS. They would also have a medium to voice concerns they might have about upgrading from their current ACDS. Evaluating this information would pinpoint the areas most important to the actual system users, versus theoretical needs conceived at the schoolhouse. To develop the best ACDS, users must play an integral role. Otherwise, developing a new ACDS without user input may not fully meet the user's needs and be a waste of time and resources.

If there were to be any type of follow-on system implementation resulting from this thesis, establishing relationships with the users would be a key factor in guaranteeing system integration, acceptance and ultimately user satisfaction. If the user believes he/she played a part in the development of the new ACDS, he/she will be more likely to use it.

A benefit of the survey would be the acquisition of information regarding other viable ACDSs already in existence. This would be in the form of library software systems that were designed in and for the Navy. By locating such programs, the aforementioned problems could effectively be resolved by simply transferring a copy of a proven ACDS to each command and replacing CDCS completely. On the other hand, if commercially developed software was being used and satisfying the users' needs, that would be reflected

in the survey results as well. Instead of re-inventing the wheel, the intent here would be to determine if a low-cost commercially developed ACDS was available to the Navy.

As previously stated, the results of the survey would exhibit the state of ACDSs in the Navy, thereby providing a definitive direction for this thesis. Essentially, there were three options: take no action, inform commands of an existing solution, or develop and implement a new solution. Taking no action would indicate that the current ACDSs are satisfactory. If an alternative and more effective system was found, resulting from the survey, that information would be passed to the interested library administrators. Finally, if the results of the survey indicated that a change was needed to improve the current ACDSs, LDS would be upgraded and serve as the solution.

B. COMMANDS SELECTED FOR THE SURVEY

The basic idea was to find libraries throughout the Navy that would benefit from using an ACDS. The only constraint placed on the libraries to be chosen was the size of their document collection. Libraries with 500 to 50,000 documents were targeted. Although, libraries with less than 500 would benefit from an ACDS, it is not essential for maintaining their collection. Libraries with 50,000 or more documents, typically have a budget that coincides with their collection, therefore, they can afford to purchase a large commercially developed ACDS and the proprietary hardware required.

Locating the libraries that fit the constraints defined was more difficult than anticipated. The first step was to contact the Librarian of the Navy and attempt to acquire

a list of candidate libraries for the survey. However, no such list exists, and the Librarian of the Navy indicated that this was due to their secure nature. The next approach was to contact the classified document producers and request a listing of the various client libraries they serve. Again, there was no list available. Finally, after contacting the Naval Security Group in Washington, D.C., a CDCS customer list was obtained which provided the target libraries needed. Out of the several hundred libraries listed, 75 were phoned and asked to participate in the survey, 25 of which agreed to participate. Those that did not participate were either unreachable due to incorrect phone numbers, unavailable during the survey time-frame, or had a satisfactory ACDS in place and not interested.

Since the CDCS libraries are highly specialized and secure libraries, finding a flexible ACDS to fulfill their requirements would naturally fulfill the requirements of less secure libraries. If LDS became the new ACDS of choice, its current version was already compatible with GENSER libraries. To make LDS's application generic and adaptable to more secure libraries, those libraries needed to be involved. This philosophy would permit a wide range of libraries, from unclassified to the most secure, to utilize the upgraded version of LDS.

When CDCS's contract expired in January 1994, the Navy's SSO Resource Manager decided not to renew it in lieu of user's apparent lack of enthusiasm for the program. This decision would effectively leave CDCS users to fend for themselves. With no further funding for CDCS development or support, transitioning to LDS or another ACDS would not be too difficult for current CDCS users. This being the case, the current CDCS users

would also be more willing to share their insight and ideas on how a dependable user-friendly ACDS would operate.

The survey was delivered to the local commands and the remaining were FAXED. A deadline of one and a half weeks was given to induce the participants to complete the survey and FAX it back. A dedicated phone line and 24 hour computer system were setup to retrieve the FAXES.

C. SCOPE OF SURVEY

The survey was unclassified and designed to cover several areas; a basic description of the library, its approximate collection size, the number of patrons served daily, what types of computer hardware, operating system, network and printers were being used, if any ACDS software was currently being used and what type, and several questions about desired feature and interest toward using LDS. The two page survey had a total of 20 questions and did not require a great deal of their time. (see Appendix A for survey)

D. SURVEY RESULTS

Within two weeks, 20 of the 25 surveys were returned and analyzed. The results showed that the average library had approximately 1200 documents, served from 12 to 44 patrons daily. The majority of computers were 386s and had MS-DOS 5.0 as their primary operating system. The vast majority were using CDCS of which over half indicated they were not pleased with its operation. Among the most significant complaints about CDCS were its lack of features and non-user friendly interface. In the final analysis,

19 to 1 said they would be interested in upgrading from CDCS to a new ACDS. No other ACDSs, besides LDS, were identified as potential candidates to immediately replace CDCS, because of its continued support by the Navy.

E. ALTERNATIVE SYSTEMS

The next step in developing an upgrade plan for LDS was to look at other library automation systems. Several large libraries (50,000+ documents) were contacted and queried about their ACDS. The Naval Postgraduate School (NPS) was eagerly awaiting the arrival of its new ACDS, a UNIX-based multi-user product called STILAS developed by Sirsi Corporation. The Defense Intelligence Agency (DIA) employs a DOS-based multi-user program called MAXCESS by Maxcess Library Systems, Inc. Goodfellow Air Force Base was waiting for its new system to come on-line, a VMS-based multi-user program called Galaxy by Gaylord Information Systems. And finally one of the original 75 commands contacted for the survey, the Navy Research Laboratory at the Stennis Space Center in Mississippi (MS?), had contracted out to Sverdrup Technology for its own DOS-based, multi-user ACDS in 1989, called the Mailroom Inventory System.

A demonstration version of each of the software products was requested for evaluation, including CDCS. Only three of the requested demonstration packages were available: MAXCESS, CDCS, and the Mailroom Inventory System. Because of their operating system and hardware requirements, STILAS and Galaxy came in the form of literature.

push buttons. The most significant features lacking in MIS are the search and check-in/out capabilities. Nonetheless, MIS does an excellent job of tracking documents and is relatively easy to learn and use.

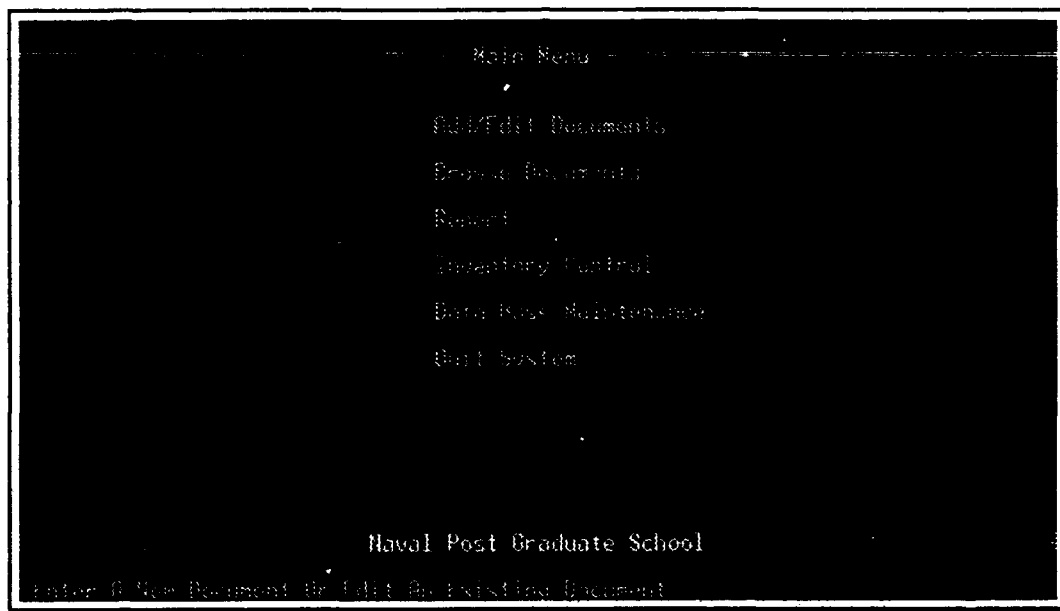


Figure 2-2 Mailroom Inventory System

CDCS (Figure 2-3) was the least preferred of the systems evaluated. Its archaic interface coupled with difficult to use features made CDCS extremely non-user friendly. CDCS 1.0 originally written by a contractor at Unisys became available in 1990. An additional \$150,000 was spent to upgrade CDCS to version 2.0 from August 1992 to January 1994. The upgrade done by Charles Fuentes at Fuentes Systems Concepts, Inc., essentially fixed a few bugs and provided a user's manual for CDCS. As previously indicated, it was mounting user complaints and dissatisfaction that caused the contract to be terminated.

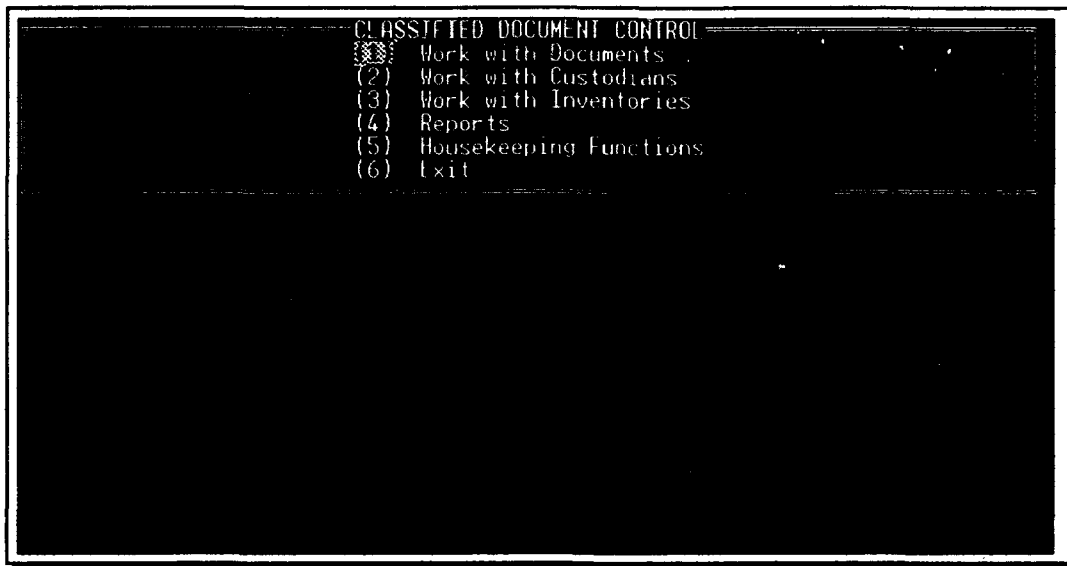


Figure 2-3 Classified Document Control System

The STILAS and GALAXY systems were not even considered. Although they incorporated many of the same features and modules as MAXCESS, both systems require end users to purchase new hardware and learn a new operating system. STILAS operates in the UNIX environment and GALAXY requires the Digital's VMS operating system. The software for these systems was very expensive not to mention the hardware investment required as well. These systems have their market niche in the large libraries with 50,000+ documents.

In summary of the alternative ACDSs, none of the reviewed systems fulfilled the requirements of small classified libraries. MAXCESS, STILAS and Galaxy were not only too large but too expensive. CDCS and Mailroom have out-dated interfaces and lack features user want.

F. UPGRADE LDS

Based on the outcome of the survey, the decision was made to move forward with the LDS upgrade. Its five year proven track record and the author's familiarity with the system made LDS the best solution. The author's experience gained by developing the original version teamed with an education in Computer Science at the Naval Postgraduate School, formed an excellent foundation from which to create the next generation of LDS.

The original version of LDS was completed in November 1989. Since then two new versions were created, NATO and Generic, to support different types of commands. Although the three versions have the same basic functionality, both the NATO and Generic versions have the capability of associating the individual's name and workspace phone number with the documents they were checking out. The NATO version has the NATO classification suite that is used by ship board and overseas libraries and the Generic version of LDS was designed for GENSER shore-based libraries in the United States. The NMITC version did not have the name association and relied on the social security number of the individual for tracking purposes, it also has ten unique barcodes that could be used for specialized classified material.

The major features of the original LDS are: the ability to add, edit, and delete documents, inventory control, reports generation, searching, check-in/out, backup and restore utilities. The LDS software is LAN ready. The user-interface is relatively simple

to use, relying on a single keystroke of the selected item number, letter, Return or Escape key to maneuver between menus or options.

The limitations of LDS like its competitors are many. From the beginning LDS was not a well planned system. During program development, modules were added as they became necessary or desired. Maintainability was difficult, especially when dealing with the three versions, due to the minimal amount of code sharing. System implementation deadlines drove the program development, which compromised structured programming design. The results were inefficient algorithms, poor variable naming, the lack of code documentation, and the GENSER classifications were inflexible. Although the LDS user interface is relatively simple, first time users found it difficult to understand and manipulate.

G. UPGRADE FUNCTIONS AND STRUCTURE

Upgrading LDS with new features and enhanced flexibility would require a significant amount of new code and restructuring. While the original code was operational, many of the procedural algorithms needed to be revamped with an assortment of functions. Using functions provides a great deal of flexibility in program design, since many functions can be shared or re-used. Variables defined locally to the function, rather than globally or privately, uses the computer's memory more efficiently. Another benefit of implementing functions was the ability to employ recursion in the search algorithm design. (see Chapter IV for more information on search) Lastly, extensive time was spent

renaming and redefining variables, program filenames and database files to permit future maintainability.

III. LDS SCHEMA IN DETAIL

A. DATABASE MANAGEMENT TECHNIQUES

For complex database systems, the importance of proper schema design can not be understated. A poor schema design wastes storage space, reduces program efficiency, and can be plagued by insert, update or deletion anomalies. Once defined, the database schema is not expected to change very often. Since LDS was being upgraded, its original database schema needed to be altered to handle the new features and additional data manipulation requirements. This process required a thorough analysis of the original schema constructs, a complete understanding of CDCS's schema make-up, and a careful examination of the new features. Two database design tools were used to aid in redefining the original LDS schema: the Entity-Relationship Diagram and the Relational Data Model.

B. ORIGINAL LDS SCHEMA

The original schema design for LDS (Figure 3-1) consisted of seven distinct database tables. Each table was developed on an ad-hoc basis to satisfy programming demands, not according to standard relational database design methods. During table construction, little attention was given to record size or redundancy of data. Certain tables were being loaded with repetitious data. Consequently, these tables could become

very large wasting valuable storage space. By employing modern relational database techniques, schema anomalies would be eliminated in the upgraded version of LDS.

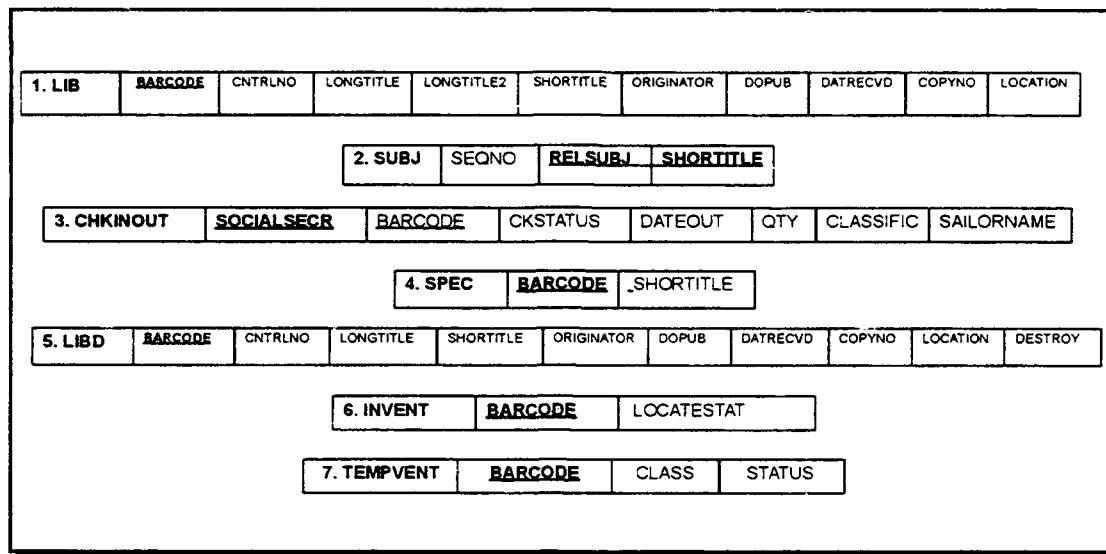


Figure 3-1 LDS Schema Design in 1989

The database tables are in the form of dBASE III PLUS .DBF files. Each table is a separate file and contains data unique unto itself, essentially making it an entity in the database system. The LIB.DBF file, for example, holds all the information pertinent to the documents in the library. The datatypes that make up each record are: character, numeric and dates.

The seven database tables relate to each other by means of primary keys and foreign keys. These keys maintain the referential integrity essential in maintaining an accurate database system. The most important key in the LDS system is the *BARCODE*. The *BARCODE* is a unique identifier for each document, just as a social security number is an unique identifier for individuals. By selecting a *BARCODE* from the CHKINOUT.DBF and relating it to the LIB.DBF, its associated document information can be obtained. This

being the case, certain situations can cause referential integrity to be compromised. In the previous example, if the *BARCODE* and its associated data no longer exist in the LIB.DBF, the referential pointer from CHKINOUT.DBF would become null. Relational database tables do not have a mechanism for avoiding this circumstance. Therefore, the LDS program had to be written to ensure and maintain the integrity of the database system.

Another issue addressed in redefining the LDS schema, was to change the field names to be more descriptive of their actual role in the table. The field *DOPUB* is an example of a field that is difficult to determine at first glance. Therefore, in the new LDS schema *DOPUB* was changed to *DATEOF PUB*. This type of renaming process will greatly assist in the maintenance of future versions of LDS.

C. CDCS SCHEMA

Since the CDCS program would effectively be replaced by LDS, it was necessary to gain an understanding of CDCS's schema make-up. CDCS was written in R:BASE™. The file structure used by R:BASE to maintain the database system's tables is completely different than dBASE III PLUS' .DBF file format. R:BASE maintains all 22 of the CDCS relational tables in three files. The only way to view and manipulate these tables, other than using CDCS, was to use the R:BASE for DOS software. Fortunately, a version of R:BASE was secured and provided a method for converting the CDCS tables into the

dBase .DBF file format. Once the tables were converted, their structure was easily viewed using dBASE III PLUS.

It was essential to provide data fields in LDS's schema that would maintain the current CDCS user's data. This process was accomplished by meticulously identifying the purpose of each field in CDCS's schema and mapping it to a corresponding field in LDS's schema. There were a few cases in which redundant data fields were either removed or combined. For example the *SYR* data field which stores the system year, e.g. 94 in each record, was removed. Analyzing the *SYR*'s function in the schema and discussing its effects with current CDCS users, indicated that many problems have occurred at the start of each year when entering new documents. Users, in several cases, would have to completely re-enter the documents from the previous year in order to get the system operational again. Another example is the *DOCID* data field. This field, originally 15 numeric characters was reduced to seven. The CDCS program required users to insert a command identification code of up to six letters. This code would automatically appear before *DOCID* numbers. The problem here was that users were required to type the command identification code with a space between it and the *DOCID* number each time they wanted to edit or view an existing document, e.g. **NPS_CS 198343**. However, the command identification code did not appear on any reports generated by the system. Therefore, the users were required to figure out when the code was required and when it was not. This inconsistency was completely eliminated in the LDS upgrade.

D. NEW LDS SCHEMA

With the schema examination of the original LDS and CDCS complete, research began into the schema requirements for the new features. This process involved documenting each new feature and creating several examples of how they would be used in the final program. Not only did this process assist in modifying the schema, but it also augmented program development.

An Entity-Relationship (ER) Diagram was constructed, once the new entities and their relationships were determined [Ref. 3]. This abbreviated ER Diagram (Figure 3-2) displays the entities as rectangular boxes, e.g. DOCUMENT, CUSTDIAN, and REPORTS. The relationships are shown in diamond-shaped boxes attached by lines to their corresponding entity types, e.g. *Have*, *Add / Edit* and *Tracks Documents*. For the previous entities and their relationships, the cardinality ratios are either 1:1 or 1:M (one to many), e.g. DOCUMENT:CLASSES in *Have* is 1:1, since each document can have only one classification. In the case where the cardinality ratio between entities and their relationships are M:N, the tables are depicted in large diamond-shaped boxes, e.g. CHKINOUT and SUBJECTS.

Through the use of the ER Diagram (Figure 3-2), two tables from the original LDS schema were converted to relationships that store data. The SUBJECTS table, for example, was identified as a relationship versus an entity because of the amount of redundant data it stored. The role of the SUBJECT table is to maintain a list of related

subjects or keywords that assist users in searching for documents. A document called "The Rise and Fall of the Soviet Union" would perhaps have related subjects: "USSR", "Soviet Union", "Military Power", etc. Another document about the former Soviet Union might also have the same keywords. In the original LDS schema the additional keywords would also be inserted into the SUBJECTS table, thus creating redundant data. The ER Diagram aided in pin-pointing this situation. As a result, the decision was made to create a new table, KEYWORDS, whose sole purpose is to maintain each unique occurrence of keywords.

The keywords maintained in the KEYWORDS table can be related to all applicable documents as necessary. The creation of the KEYWORDS table eliminates redundant data and wasted storage space. Since keywords are entered only once the opportunity for data entry errors are significantly reduced as well. From the previous example using the old LDS schema, for each new document entered into the system, the librarian would have to enter all the applicable keywords. In the event the librarian typed "USST" instead of "USSR", a patron's search for "USSR" would not reveal the miss-typed keyword and its associated document. Although strict attention to keyword entry would help in avoiding the problem, the new LDS schema and program displays the current list of keywords available in the system for the librarian to choose from while entering a new document. The keyword algorithm allows the librarian to type the first few letters of the keyword he/she intends to enter. It then attempts to find a match in the current list of keywords. If one Soviet document was previously entered the keyword list would display "USSR" as an

optional keyword. The librarian would then select "USSR" or continue to type a new keyword into the system.

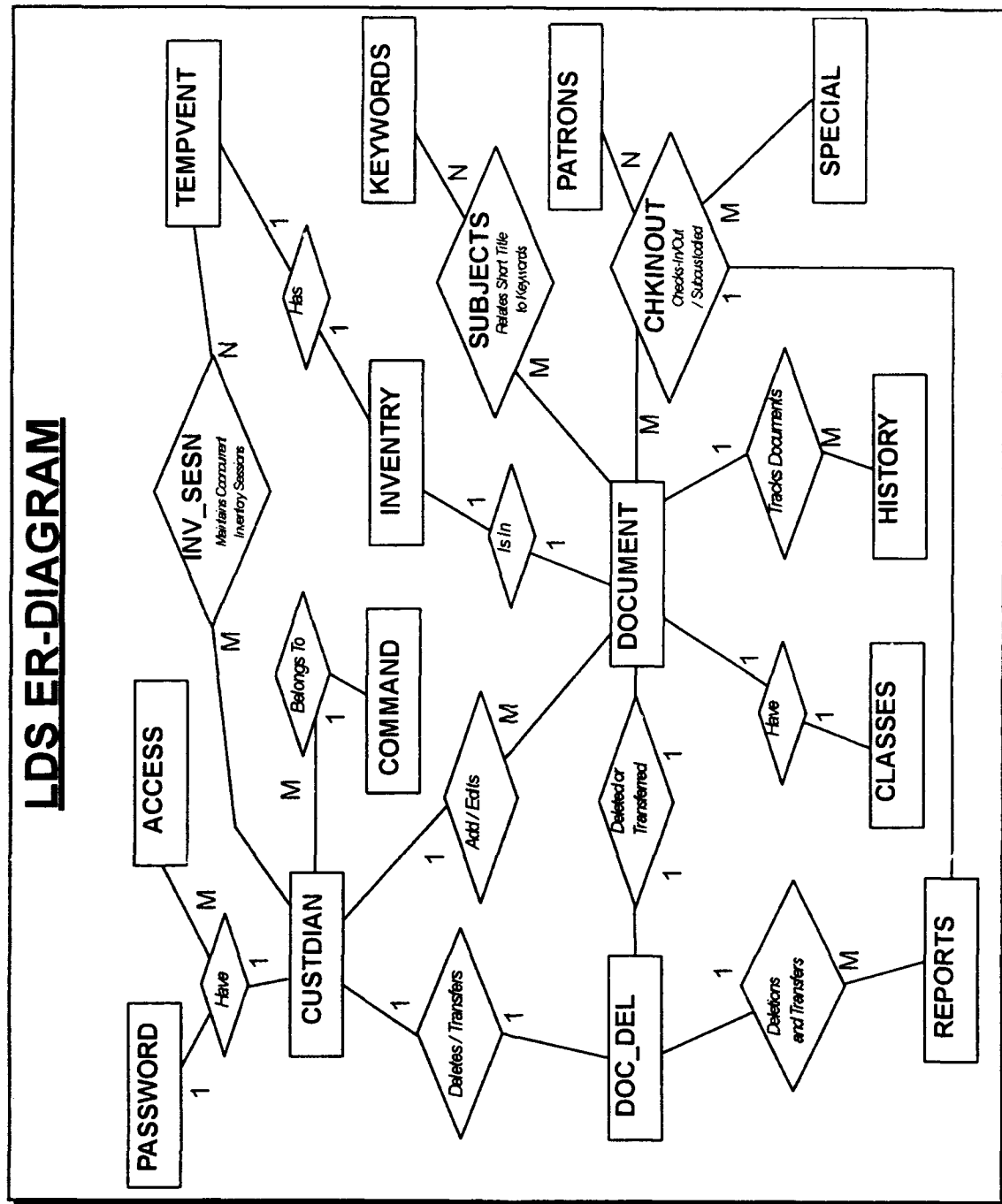


Figure 3-2 LDS Entity-Relationship Diagram

The benefits of constructing an ER Diagram were many. It simplified the addition of the 10 new tables to the LDS schema by reducing the complexities between entities and relationships. Even in its abbreviated form, the ER Diagram provides an overall understanding of how the system operates. Finally and most importantly, the ER Diagram revealed potential problems in the schema design. The resolution of these obstacles was much easier in the design phase than during the system implementation phase.

Prior to the actual schema implementation, the new LDS tables were outlined using the Relational Data (RD) Model (Figure 3-3). This enabled the characteristics of each relation to be defined in more detail than the ER Diagram permits. The new attributes were defined and named according to their relation's role in LDS, e.g. in the HISTORY table, *BARCODE* becomes *HBarcode*. The primary or original LDS relation's retained their basic data field names, for example, the DOCUMENT.DBF which is a primary relation retained the data field *BARCODE* as *Barcode*. This was done to disambiguate the primary and secondary or subordinate relations when upgrading the LDS program. Since the previous version of the LDS program was written using the primary tables and their respective attributes, adding new tables would be accomplished more easily by employing relation specific attribute names.

The RD Model was also used to define the key integrity constraints for each table. In each of the 17 tables, the primary keys are shown in bold and underlined (Figure 3-3). The foreign keys have not been labeled for simplification purposes, but were documented while constructing the RD Model.

1. LIB	<u>Barcode</u>	Control_No	Longtitle	Shorttitle	Originator	DateOfPub	DateRecvd	CopyNumb	Location	Command_ID	Class_Numb
	Status	Custodian	Defcos	Notes	Dispo_Date						
2. SUBJ	<u>Subj_Numb</u>	<u>Shorttitle</u>	<u>Key_Number</u>				3. KEYWORDS	<u>Key_Number</u>	Keyword		
4. CUSTDIAN	<u>CSSN</u>	CLastName	CFirstName	Cust_ID	CRate_Rank	CLocation	CPhone	CDateInput			
	CType	CAccess									
5. PASSWORD	<u>Cust_ID</u>	Keyword				6. ACCESS	<u>Access_SSN</u>	Acc_Levels			
7. CHKINOUT	<u>SSN</u>	Barcode	CkStatus	DateOut	Quantity	Classific	Chk_Out	Custodian			
8. SPEC	<u>BARCODE</u>	<u>SHORTTITLE</u>				9. REPORTS	<u>Report_Num</u>	RCustodian	RDate	RTime	
10. DOC_DEL	<u>Barcode</u>	Control_No	Longtitle	Shorttitle	Originator	DateOfPub	DateRecvd	CopyNumb	Location	Command_ID	
	Status	Custodian	Defcos	Notes	Dispo_Date	Report_Num	Class_Numb				
11. COMMAND	<u>Comand_ID</u>	Comd_Name	Destroy_YR	Authorized	Auth_Off	Destr_Off	Witness_Off	Disk_Bkups	Printer		
	Enter_User	Purge_User	Special	Title_Clas	Security						
12. HISTORY	<u>HBarcode</u>	HTrans_Act	HDate	HTime	HCustodian						
13. INVENTORY	<u>BARCODE</u>	Class_Numb	Location	Status				14. TEMPVENT	<u>Barcode</u>		
15. INV_SESN	<u>Cust_ID</u>	Type	Describe	Start	Stop						
16. PATRONS	<u>PSSN</u>	PLastName	PFirstName	PRate_Rank	PLocation	PPhone	PDateInput	PAccess			
	PExpires										
17. CLASSES	<u>Class_Numb</u>	Class_Abrv	Class_Def								

Figure 3-3 New LDS Relational Database Schema

Enhancing the RD Model with data types, attribute field size and related indexing information provided the structure necessary to begin creation and modification of the LDS tables (see Appendix B). The ER Diagram, RD Model, and the enhanced RD Model provided the means for generating a sound schema design and moving forward with program development and implementation.

IV. SEARCH

A. JUSTIFICATION FOR SEARCH

Automated classified document systems (ACDS) that simply track documents are nothing more than glorified handwritten log-books, such as the Mailroom Inventory System and CDCS. Although the speed and legibility factors are vastly improved, how do these "basic" tracking systems assist patrons in locating the documents they need? Documents sitting on the shelf are practically useless if the information within can not be accessed easily. With a "basic" ACDS, there are only a few ways for patrons to access a library's collection. One method is to allow patrons to literally thumb through the documents. This method can be frustrating and time consuming. The patron will eventually find what he/she needs or give up. The librarian can be an another resource for finding information. However, this requires the patron to query the librarian until the desired documents are retrieved. Although this method can be effective, it too can be lengthy and inefficient for both parties.

The ability to conduct complex searches of a database is a primary function of commercial ACDS, such as MAXCESS, STILAS and GALAXY. After considering these circumstances, it became evident that a search feature should be a required component of an ACDS. Therefore, the decision was made to upgrade LDS with a search feature that would provide immediate and informative access to a library's collection.

B. PREVIOUS SEARCH

Even though some level of search capability is preferable to none, the original LDS's search feature has some limitations. Its most capable search, the "Two Subject Search" (Figure 4-1), is essentially a Boolean *AND* search of two keywords. This is not a serious constraint, until the patron or librarian must type the two keywords exactly in order to begin the search. If a syntax error or misspelling occurs, the user is prompted to start over. Even with a print-out of the keywords, typographical errors are inevitable and lead to user frustration. Furthermore, since there is not an easy way for users to view the current keywords in the system, keyword print-outs potentially contained out-of-date information.

The addition of new documents and their associated keywords into the system can be another source of problems. Unless a list of standardized keywords is available when librarians enter documents, the resulting keywords can be left to the librarian's interpretation. For example, if a document being entered pertained to the former Soviet Union, one librarian might enter "*USSR*" as a keyword, another librarian given a similar document might enter "*U.S.S.R.*", or "*Soviet Union*", or "*Russia.*" Therefore, since the keyword "*U.S.S.R.*" is not the same as "*USSR*" in the system, a user searching by the keyword "*USSR*" would not receive any data associated with the "*U.S.S.R.*" keyword.

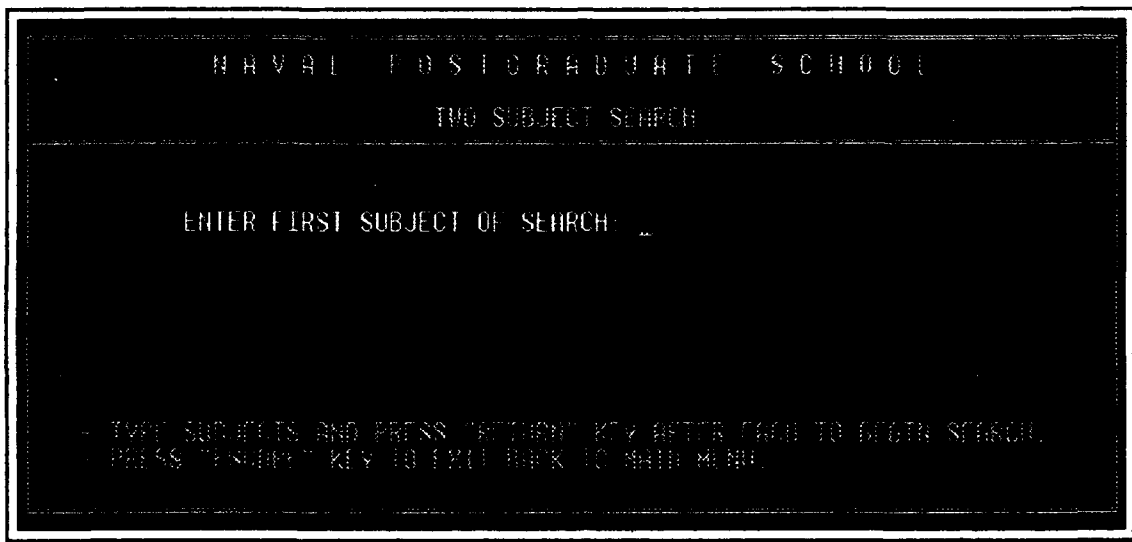


Figure 4-1 LDS '89 Two Subject Search Feature

When a search was successful, the resulting screen display consisted of library specific information, e.g. barcode, control number, and the document's title. From the title, the users would decide whether or not they wanted the document, no further information was available. The user had no way of knowing if the document was checked-out or whether its contents were more or less technical than they needed. Furthermore, if the user wanted to conduct a follow-up search on the information displayed, he/she was required by the system to re-enter the keywords. Here again the original LDS search feature was limited.

Even with the restrictions described above, the search feature in LDS was a useful tool, a tool noticeably missing from CDCSy

and the Mailroom Inventory System. Given LDS's scope and its background, creating a flexible yet robust search would be challenging.

C. NEW SEARCH OBJECTIVES

The emphasis in planning the new search for LDS was placed on creating a flexible Boolean search capability. Ideally this capability would support *AND*, *OR*, *NOT*, and parenthesis for complex search manipulation. Since the commercial ACDS provide Boolean search mechanisms, adding a Boolean search to LDS would make it a more valuable system.

Keyword Search:

Subject:

- ☐ And
- ☐ Or

- Soviet Technology
- Soviet Union
- Soviet Union - Ships
- Soviet Union - Subs
- Soviets
- Technology - Radar
- Technology - Tests

Figure 4-2 Preliminary Design of New LDS Search

The lessons learned from the original LDS search indicated that the keywords must be readily available to users. A keyword display screen was envisioned (Figure 4-2), that would effectively eliminate typographical errors and out-of-date keywords. The users would compose their search by selecting keywords with a mouse, or by using the arrow keys to scroll through the keyword list and pressing the enter key when the desired

keywords are found. Additionally an advanced touch-type feature was conceived that would permit the users to type the first few letters of the keyword they want. This would move the cursor bar to the first keyword equaling their touch-type entry. If a match is found the users would select the keyword, otherwise they could scroll through the choices and complete their search as necessary. The same keyword display screen could be utilized by librarians when entering documents and associated keywords. This would reduce data entry errors and create a standardized set of keywords.

Once the documents matching the search criteria are found, the resulting display should enable users to access additional information on any given document. This information would be in the form of an abstract, entered by the librarian when the document is initially input into the system. The abstract would provide a more detailed description of the document's contents, than one could glean from its title. A portion of the abstract screen display would indicate whether the document was available or already checked-out. In addition to the abstract feature, users would be shown "*SEE ALSO*" keywords for further reference. By selecting the "*SEE ALSO*" feature, users would have the ability to refine their search even further.

Once the Boolean search feature is fabricated, its scope could be broadened to permit searches composed of various document elements. For example, a patron might want to find a document produced by *NPS* with keywords: *Computers AND Libraries*, authored by *Elkern*, with the word *LDS* in the title.

D. SEARCH ALGORITHM

The development of the search algorithm required several steps in order to achieve the desired flexibility of a Boolean search. Sample searches were created, using *AND* or *OR*, and inserted into symbolic search tree structures. The search tree structures were then converted into a two dimensional array format for easy manipulation. Once the step-by-step array traversal scheme was complete, a recursive search process was developed to locate the matching keywords and associated data. Although all the objectives identified for the Boolean search were not realized, it was designed for future expansion. In its current form, the upgraded LDS search is the most complicated algorithm in the system.

Several examples were compiled to simulate typical user generated searches. These examples were then input into symbolic search tree structures. For example, the user might compose a search: *Engines AND Holodeck* (Figure 4-3). In accordance with the new schema design, each keyword is assigned a unique number, e.g. *Engines* is assigned 7 and *Holodeck* is assigned 9. Similarly for the search process, the Boolean connectors *AND* and *OR* were assigned 1 and 2 respectively. By assigning character data, numeric values, random access memory (RAM) consumption was reduced and search logic was easily manipulated.

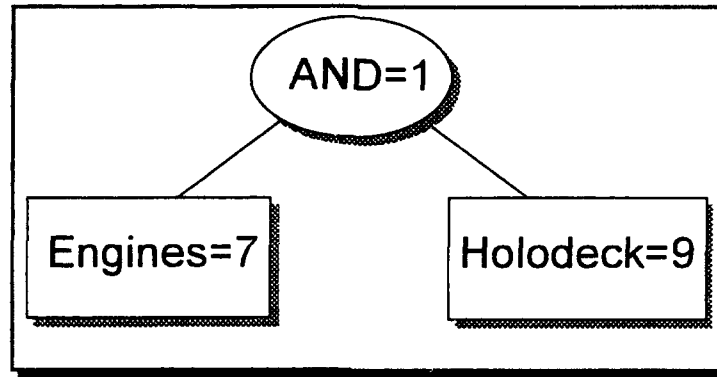


Figure 4-3 Symbolic Search Tree Structure

After constructing several examples, a two dimensional (2-D) array format was conceived to store the search connectors and keywords as the search was generated. The result was a 20 by 5 2-D array (Figure 4-4). This array design would provide ample cells for the allocation needs of the sample *AND* and *OR* searches devised.

Element	Connector	Left Index	Left Keyword	Right Index	Right Keyword
	1	2	3	4	5
1	0	20	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	1	0	7	0	9

Figure 4-4 Two Dimensional Search Array

The table header (Figure 4-4) identifies the role of each cell column. The *Element* columns and rows are for clarification purposes only. The *Connector* column contains the *AND* or *OR* connectors, 1 or 2 respectively. A break point of row 10 was made between the *ANDs* and *ORs*. Above row 10, all connectors are to be *ORs* and from row 10 and below all connectors are to be *ANDs*. The *Left Index* and *Right Index* columns contain the row numbers for the symbolic left and right side of the search tree. Lastly, the *Left Keyword* and *Right Keyword* are the locations where the keyword number are stored

according to their position in the symbolic tree structure. The example in Figure 4-3, *Engines AND Holodeck*, has been inserted into the table in Figure 4-4. Starting from row 1, column 2, the *Left Index* indicates row 20 as the current location of the search. Following the arrow to row 20, in the *Connector* column a 1 appears indicating *AND*, and both keywords appear in their respective *Left Keyword* and *Right Keyword* locations as they appeared in the symbolic tree structure (Figure 4-3).

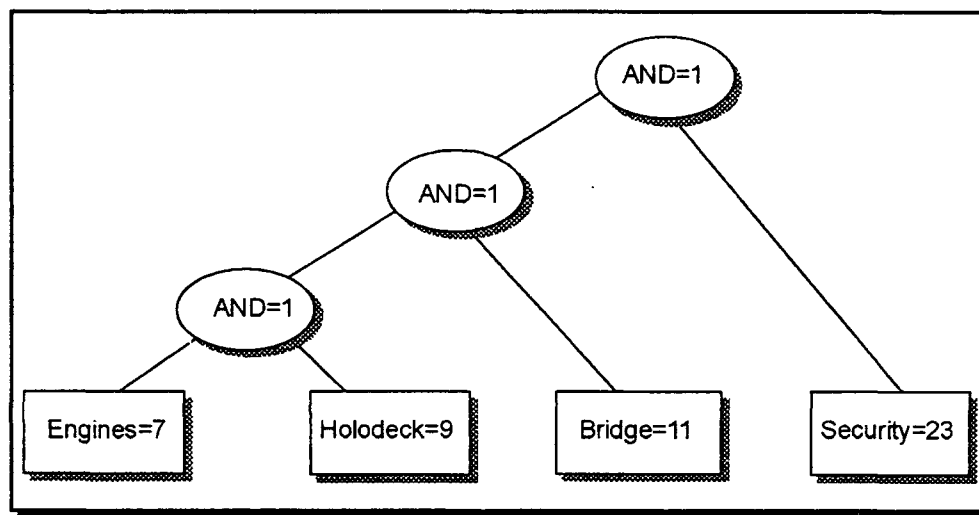


Figure 4-5 Complex AND Search Example

A more complex example would be the search for: *Engines AND Holodeck AND Bridge AND Security*, or in numeric representation: 7-1-9-1-11-1-23 (Figure 4-5). The resulting two dimensional table created can be seen in Figure 4-6. Although this is the largest search allowed in the current upgrade of LDS, it is plain to see that there is ample room for future expansion of the search feature.

Element	Connector	Left Index	Left Keyword	Right Index	Right Keyword
	1	2	3	4	5
1	0	18	0	0	0
2	0	0	0	0	0
17	0	0	0	0	0
18	1	19	0	0	23
19	1	20	0	0	11
20	1	0	7	0	9

Figure 4-6 Complex AND Search Table Result

For completeness, a search example that uses both sides of symbolic tree structure would be: *Engines AND Holodeck OR Bridge AND Security* (Figure 4-7). The resulting table indicates how the array traversal would satisfy the search request (Figure 4-8). The search algorithm was designed to handle more complex queries, however, since parenthesis were not implemented, the number of connectors a user can combine is four.

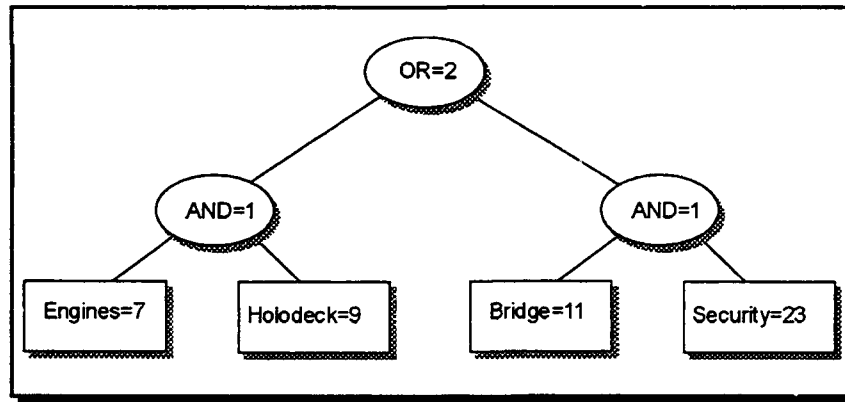


Figure 4-7 Left and Right Search Tree Example

Element	Connector	Left Index	Left Keyword	Right Index	Right Keyword
	1	2	3	4	5
1	0	20	0	19	0
2	0	0	0	0	0
18	0	0	0	0	0
19	1	0	11	0	23
20	1	0	7	0	9

Figure 4-8 Resulting Left and Right Side Search Table

With the array construction scheme operational, development began on a recursive search process that would locate the 2-D array's contents in the database. Essentially this involved traversing the 2-D array until the bounds of the search were satisfied. In terms of the symbolic tree structure (Figure 4-7), the search begins at the right and traverses toward left. Depending on the connectors found during the traversal, the data located is handled accordingly; *AND* requires intersection of previous keyword and next keyword, and *OR* requires the union of previous keyword and next keyword. As keywords are located in the database, their associated *Shortitle* (see Chapter III LDS Schema In Detail) is stored in a separate *Shortitle* array. The algorithm then calls itself recursively, inserting the next keyword into the search parameter. If the keyword is located and depending on the connector between the current keywords (*OR* or *AND*), its *Shortitle* is either added to the current *Shortitle* array, or it is discarded if it doesn't match any of the previous *Shortitles*. This process continues until the search is resolved.

E. NEW SEARCH FEATURES

Once the Boolean search was programmed, the user interface was designed. The goal was to incorporate as many of the stated objectives as possible, from the lessons learned based on the original LDS's search, to the requirements of implementing the Boolean search. The most significant new feature is the ability for users to view and select the keywords on screen (Figure 4-9).



Figure 4-9 New LDS Search User-Interface

There are numerous ways to view and select the keywords from the on screen list. The user can use a mouse to scroll down the list clicking on the word he/she wants, depressing on the arrow or page-up/down keys will also allow the user to scroll through the entire list of current keywords, and perhaps the most useful way of locating the desired keyword is by simply typing the keyword. A touch-type algorithm seeks the letters

one-by-one as the user types, attempting to find a match. When the match is found the cursor bar rests on the keyword for the user to select. For example, if the user types *ENG*, the first word matching the letters will be highlighted by the cursor bar (Figure 4-10). The user can then press *Return* or double-click on the word with the mouse to select the keyword.

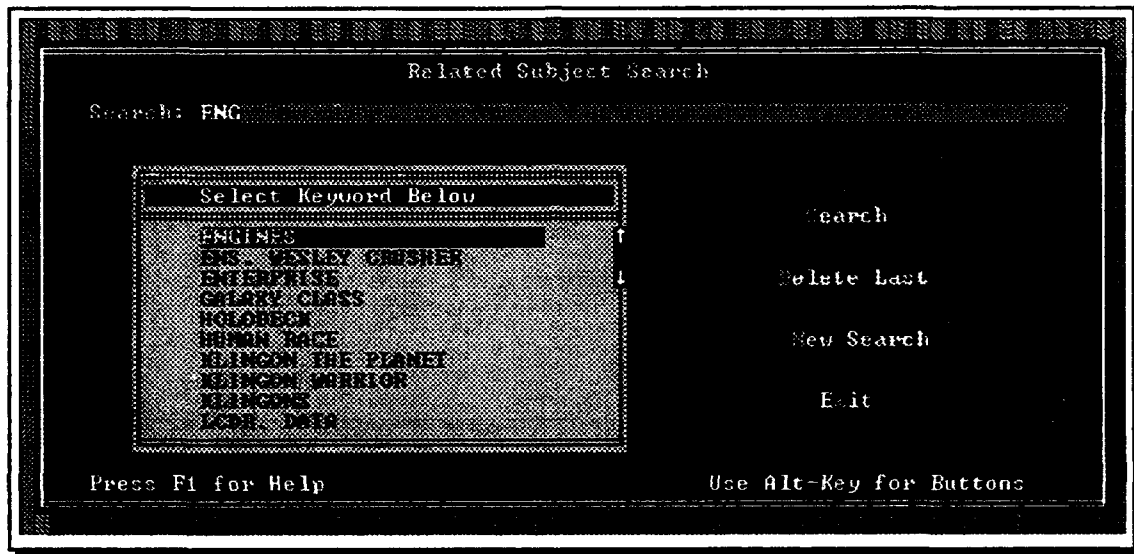


Figure 4-10 Touch-Type Example

After the first word is chosen, the Boolean search dialog box appears on the screen (Figure 4-11). At this point the user can choose to carry out the search based on what appears in the search field, or select the *AND* or *OR* buttons to create a Boolean search. (For a complete search example, see Chapter V.) By reducing the amount of typing the user must do while creating their search, potentially time consuming errors are eliminated. If the user select a Boolean connector, it is displayed in the search window and the user is once again free to select another keyword from the list. This process, keyword/connector,

continues until the user selects *SEARCH* or attempt to add more than three Boolean connectors. With the current release of LDS, users can search for up to four keywords.

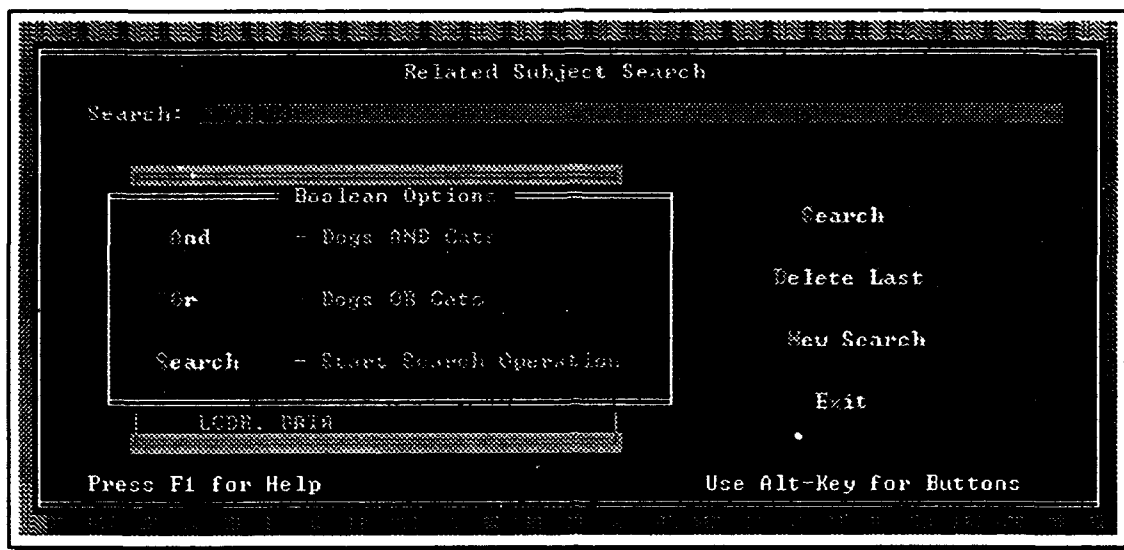


Figure 4-11 Boolean Dialog Box

After the user chooses search, if documents are found matching the search criteria, they are displayed on a subsequent search results screen (Figure 4-12). From this screen users can elect to view an abstract from any of the document displayed. The abstract not only provides more specific information on a document, but also informs the user of the documents status.

Other useful elements have been added to the search result screen, such as: the user can now view the search they input at the top of the screen, the current page and the total number pages is displayed at the bottom left corner, and when the user is finished reviewing the current results he/she can select the *SEARCH MENU* button. The *SEARCH MENU* button allows him/her to edit the previous search, add an additional constraint, delete a constraint, start a new search, or exit the search feature completely.

Search Results For: <u>ENGINE</u>		
# Control No.	Long Title	Originator
1 U-399337 411343	DATABASE MGMT SYSTEMS	DATA INC
2 U-2981344 221218	PENTIUM PROCESSOR - CISC OR RISC	INTEL
3 U-18199 111114	MAN INTERFACE	COMSCI
4 U-29864 111141	OS/2 - IBM	IBM
5 U-9889 120123	MISSION RISC CHIP	APPLE
6 U-2945 220129	INTERESTING REVIEWS OF COMPUTER MONITORS	NEC
7 U2174 111115	CHICAGO - WIN 4.0	MICROSOFT

Figure 4-12 Search Results Screen

The new search algorithm satisfies the primary goal of creating a flexible Boolean search. This will enhance the user's ability to locate desired information quickly, while reducing the reliance and burden on librarians. The abstract capability permits the user to evaluate the contents of a specific document as well as view its current status. The keyword display feature assists both in searching and adding documents as it maintains a list of standard associated keywords. Although not all the objectives for the search upgrade were implemented, this new search is far superior to the original LDS's and is comparable to search features found in commercial ACDS.

V. LDS EXAMPLE

A. BREADTH OF EXAMPLE

The new version of LDS not only contains several new features, but was virtually rewritten using the structural programming and object-oriented techniques CA-Clipper provides. This being the case, the purpose of this chapter will be to highlight the most significant features that have been incorporated into the new version of LDS. The features to be demonstrated include: Search, Classifications, Custodians, Documents, and the on-line Help facility. The data used in all examples is unclassified and completely fictitious. To acquire a true appreciation for the features previewed here, one should procure a copy of LDS.

B. SEARCH EXAMPLE

The new Boolean search implemented in LDS is very powerful and easy to use. Its attributes include: a keyword display window, touch-type keyword access, keyword scrolling and selection by using the keyboard or mouse, Boolean *AND* or *OR* search construction, and document abstract viewing (see Chapter IV Search, for more information). The following example will demonstrate how the Boolean *AND* assists in reducing the search results.

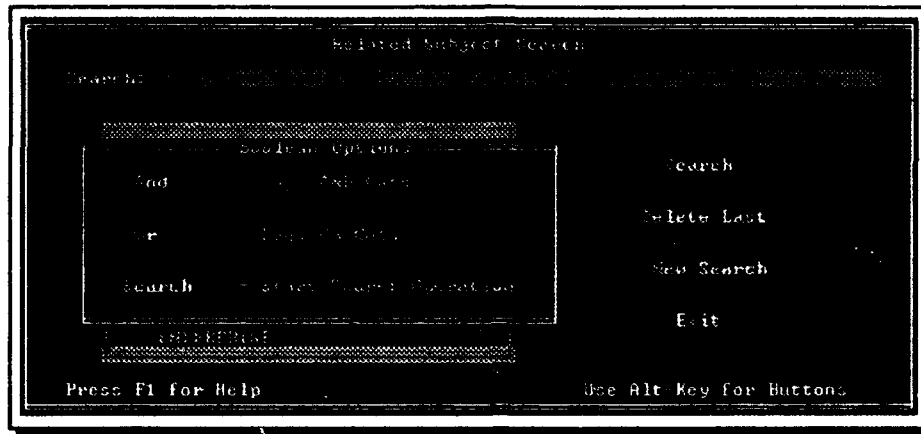


Figure 5-2 Boolean Options Dialog Box

If documents matching the search criteria are found, they are displayed on a results screen (Figure 5-3). From this screen the user can view an abstract for a specific document and its current status or return to the search menu. Abstracts are viewed by pressing **Alt-A** or by clicking on the *Abstract* button with the mouse. An abstract dialog box appears and requires the index number (# column at far left, range from 1 to 8) of the document in order to display its associated abstract (see Add/Edit Document example for sample abstract).

Search Results for: ENGINEERING			
#	Control No.	Long Title	Originator
	Short Title		
1	0-29339	DATABASE MONI SYSTEMS	DATA INC
2	0-29810344	PENTIUM PROCESSOR - CISC OR RISC	INTEL
3	0-29810344	MAN INTERFACE	COMSCI
4	0-29810344	MANIP	IBM
5	0-29810344	CS22 - IBM	IBM
6	0-29810344	ARCADE	ARCADE
7	0-29810344	MIDNIGHT RISK CHIP	ARCADE
8	0-29810344	INTERESTING REVIEW OF COMPUTER MONITORS	REC
9	0-29810344	CHICAGO - WIN 4.0	MICROSOFT

Figure 5-3 *ENGINES* Search Results

If the *Search Menu* button is selected, another dialog box appears asking the user if he/she would like to edit the previous search, start a new search or exit the search feature completely. For this example, the edit previous search option is chosen and two additional keywords are added connected by Boolean *ANDs*, forming the search string: *ENGINES AND SECURITY AND BRIDGE OF USS ENTERPRISE* (Figure 5-4). If at any time during the search composition stage an incorrect keyword or Boolean connector is selected, the user can push the *Delete Last* button and remove the last item entered. Since Boolean *ANDs* are used, the search results will be reduced from those displayed in Figure 5-3.

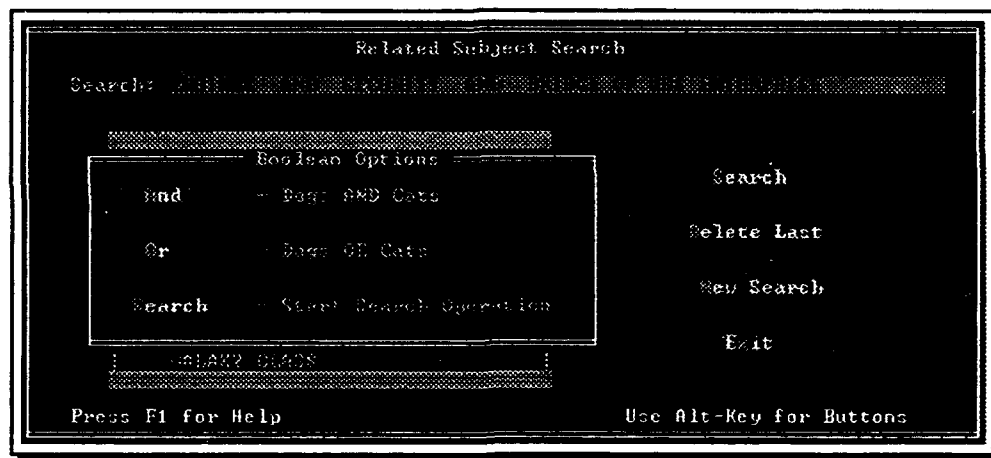


Figure 5-4 Modified Boolean Search

After choosing the *Search* button with the mouse, the search begins again, this time with the Boolean constraints placed on the resulting documents. When the results are compiled the document display screen appears with the matching documents a fraction of those found when *ENGINES* was searched for previously (Figure 5-5). The same options

that were available after the completion of the first search, are also available on the document display screen in Figure 5-5.

Search Results for: MISSION RISC CHIP INTERESTING REVIEWS OF COMPUTER MONITORS			
#	Control No.	Long title	Originator
	Accession	Short title	
1	0-8989	MISSION RISC CHIP	APPLE
	000-03	RISC-CHIP	
2	0-2945	INTERESTING REVIEWS OF COMPUTER MONITORS	NEC
	000-03	OVERVIEW	
Page: 1 of 1 Abstract Search Menu			

Figure 5-5 Results of Second Boolean Search

C. CLASSIFICATIONS

The incorporation of a flexible classification feature in the new LDS was paramount. Although LDS will have a set of default classifications defined as defaults, the users will have the ability to create, modify and delete classifications at will.

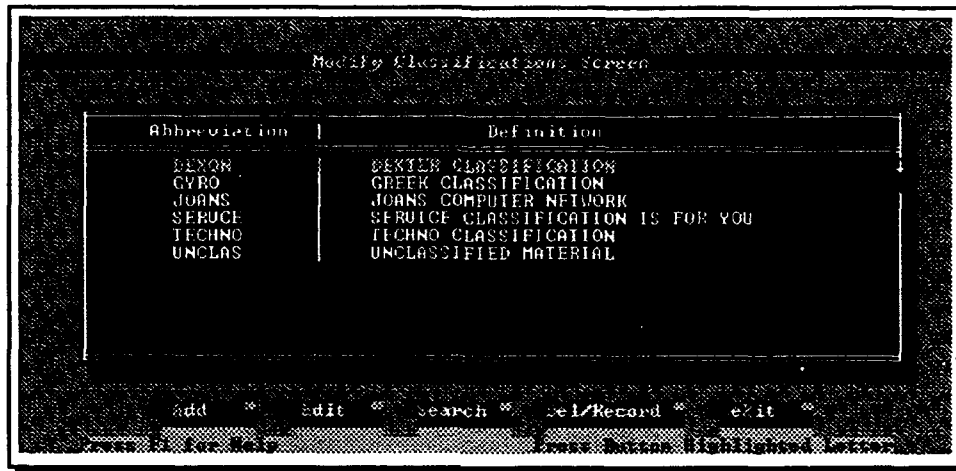


Figure 5-6 Classification Interface

The classification interface was designed to be user friendly and consistent with other screen layouts in the system (Figure 5-6). The existing classifications are found in the window centered on the screen. The abbreviation column (left) is where each abbreviation appears for its corresponding definition (up to 40 characters) of the classification in the right column. The buttons at the bottom of the screen can be accessed by either pressing the highlighted letter, e.g. *A* for *ADD* or by clicking on the button with the mouse.

The classification *DESKTOP IV CONTRACT* will be input for this example. The first step is to create a six character abbreviation for *DESKTOP IV CONTRACT*, e.g.

DSKTP4. By pressing the *ADD* button with the mouse a classification input dialog box appears in the center of the screen (Figure 5-7). Note that the buttons at the bottom of the screen temporarily disappear while the new classification is being added.

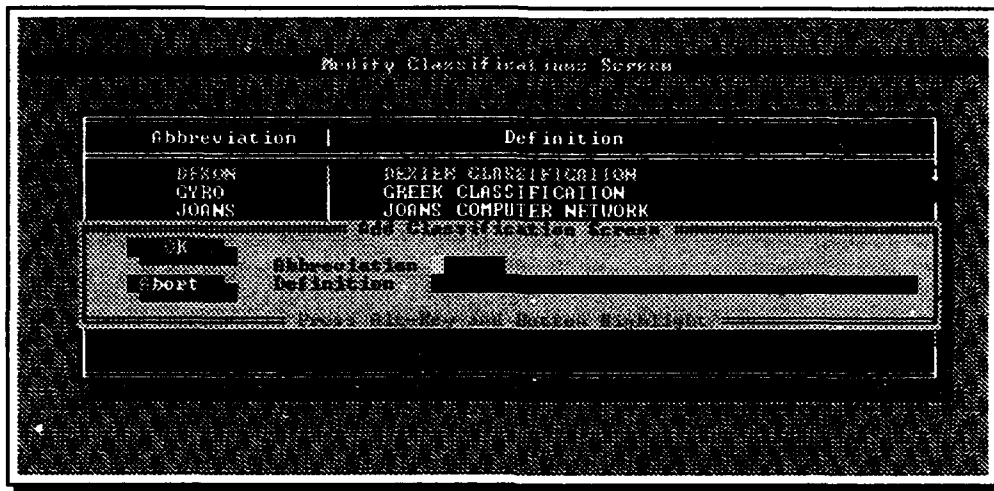


Figure 5-7 Add Classification Dialog Box

Once *DESKTOP IV CONTRACT* is typed in, choosing the *Okay* button causes a new dialog box to appear which permits the user to verify his/her entry (Figure 5-8). If *DESKTOP IV CONTRACT* or its abbreviation was miss-typed, this dialog box would allow the user to modify the new classification or cancel its entry altogether. By pressing *Return* the new classification is added to the current list of classifications. The edit feature operates the same as the add, simply choose the classification to edit and press the edit button with the mouse.

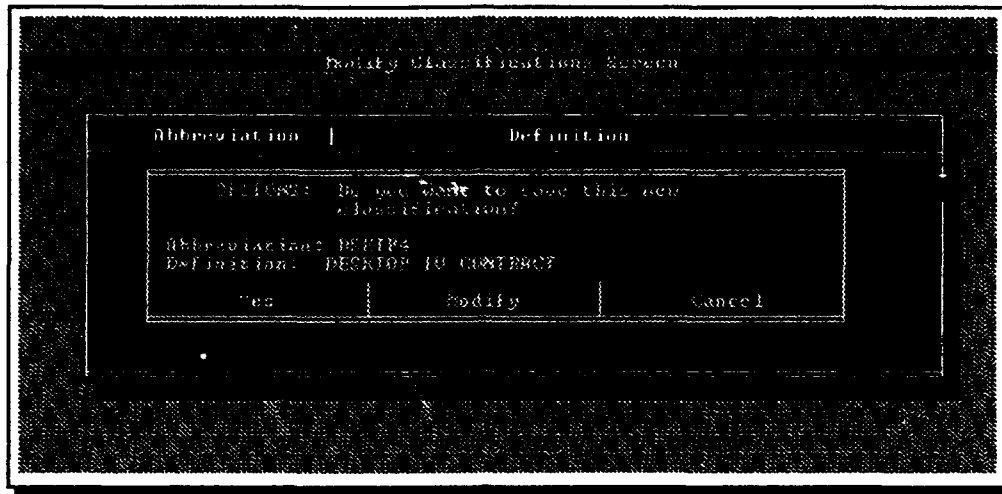


Figure 5-8 Classification Verification Dialog Box

The classification search is particularly useful for commands that have dozens of classifications. When activated, the search option moves the cursor bar to the location of the classification abbreviation entered by the user. For example, to search for the *DSKTP4* classification, the search option is activated by pressing the search button with the mouse or by pressing the letter *S*. A prompt appears where the abbreviation is to be entered (Figure 5-9). This feature is very similar to the touch-type feature found in the Boolean search component, in that as the user types the abbreviation, the cursor bar moves through the list of abbreviations to the one that matches. If no match is found a low pitch tone sounds to indicate search failure. Once the classification is located it can be edited or deleted.

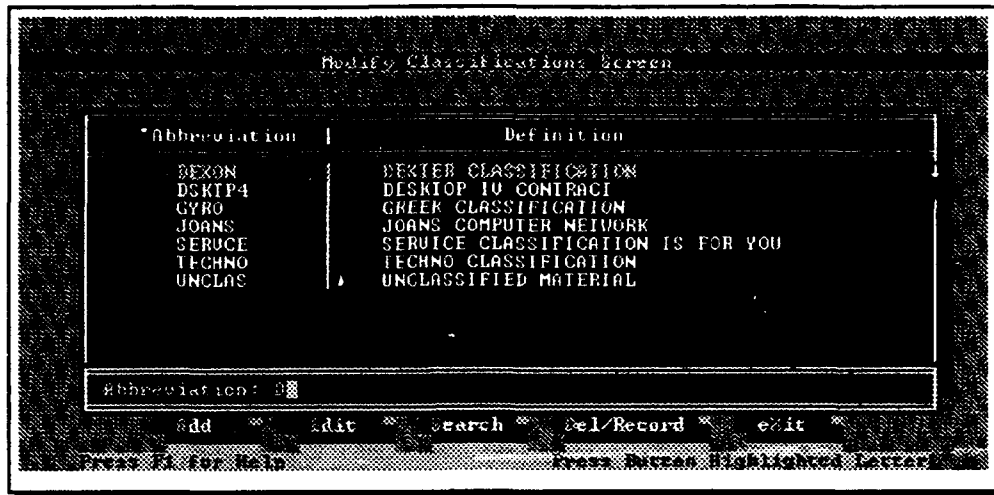


Figure 5-9 Classification Search

D. CUSTODIANS

Since custodians play such a critical role in the management of classified documents, the new version of LDS facilitates the addition and maintenance of custodians. Consistent with the previous classification feature, the custodian maintenance interface was designed to be user friendly (Figure 5-10). Once entered into the system, custodians can be assigned to specific documents indicating their accountability for those documents.

Custodian Maintenance Screen					
Cust ID	Rank	Last Name	Location	Phone	Type
06NAD6	CIO	ANDREWS	UA	(800)115-4874	S
ELKERN94	LI	ELKERN	NAVAL POSTGRADU	(408)656-2124	P
JIM98.	LI	JENKINS	UBOAT	() 0	S
2584	LI	LOBBETT	WPS	(202)215-4355	P
STEIN94	CIO	STEINBECK	TOP GUN	(408)558-7884	P
J8844	LI	SVANSON	UPSTAIRS ROOM 2	(800)212-1188	S

Add Edit Search Del/Record Exit
 Press F1 for Help Press F2 for Highlighted Letter

Figure 5-10 Custodian Maintenance Interface

Adding a new custodian consists of pressing the *Add* button and filling the corresponding data fields with the individual's social security number, custodian ID, name, etc. For example to enter a new custodian by the name of *Lt. Tom Wilson*, the *Add Custodian Screen* would appear as Figure 5-11. After entering the new custodian the user is asked to input the custodian's password. The user must enter the password twice to verify its correct spelling. If an error occurs during password entry, the user is prompted to start over. The password feature is useful in areas of LDS that require proper authorization to function, such as deleting documents. The *Edit*, *Search* and *Del/Record* functions operate just as those in the Classifications feature described above.

Custodian Maintenance Screen

Cust ID	Rank	Last Name	Location	Phone	Type
SEAN8DR	CI			4824	S
LLKLRN94	LI			2174	P
JIM98	LI			0	S
1554	LI			4455	P
STEIN94	CI			7884	P
J8844	LI			1188	S

OK

short

Site: 388 88 8888

Cust ID: MILSON

First Name: THOMAS

Last Name: MILSON

Rank Rate: LI

Location: CUIC USS WASHINGTON

Phone: () -7494

Type: SUB CUSTODIAN

Status: COMPLETE

Date Input: 7409205

Press Alt Key and cursor highlight

Figure 5-11 Add Custodian Screen Example

E. DOCUMENTS

The *Add Document* feature provides for the entry of all the characteristics which describe classified documents. In this new version of LDS, 11 new data fields have been incorporated for a total of 18. The most significant of these new data fields are: *Classified*, *Custodian* and *Abstract*. The overall interface layout is consistent with those of the *Classifications* and *Custodians* features (Figure 5-12). The new elements of *Add Documents* and its interface operation will be explained in the following example.

Add Document									
Barcode	Control No	Short Title		Classified					
1301	1301	Data Junction							
Long Title		(press TAB key to expand)							
Original Pub Agency	Date of Pub	Date Received	Appr Date	Review Date	Disposition Date				
	94/06/08	94/09/06	0	0	/ /				
Location	Custodian	Status	Custodian	Abstract					
	13	ACTIVE		(press TAB key to expand)					
Remarks: N/A		Abstract							
New Custodians Exit									

Figure 5-12 Add Document Interface

The document to be added in this example is called "*Data Junction*." The *Barcode*, *Control No* (Number) and *Short Title* are entered first. Of the 18 data fields, the two most critical fields are the *Barcode* and *Short Title*. Both fields must be entered before moving to the *Classified* data field. If the user attempts to by-pass either of these two fields an error message will appear accompanied by a warning tone. The example *Barcode 1301* is entered, and by pressing the *Return* key, the cursor moves to the next field, *Control No*.

A *Control No* of U-28094 is entered, and the *Short Title* is "Dat-Junc-94." Pressing the *Return* key once again at the *Short Title* field, moves the cursor to the *Classified* field and pops-up a listing of all the currently available classifications in the system for the user to select (Figure 5-13). The benefit of this pop-up window is that if the user does not remember the abbreviation for a certain, he/she can scroll up or down the list to find the classification he/she wants. Additionally the touch-type feature described in the Boolean search works here as well to aid in finding the correct classification abbreviation. The classification is selected by pressing the *Return* key. The six letter abbreviation is placed in the *Classified* data field.

The screenshot shows a terminal window titled "Add Document". It contains a form with the following fields: "Control No" (1301), "Short Title" (U-28094), and "Classified" (DAT-JUNC-94). Below these fields is a "Pop Up Classification Screen" with a table of classifications. The table has three columns: "Long Title", "Abbreviation", and "Definition". The "Long Title" field is currently empty. The "Abbreviation" field contains a list of abbreviations: DEXON, DSKIP4, GYRO, JOANS, SERUCE, TECHNO, and UNCLAS. The "Definition" field contains the corresponding definitions for each abbreviation. At the bottom of the screen, there is a status bar with the text "Highlight Choice & Press Enter" and "typed TAB to expand".

Long Title	Abbreviation	Definition
	DEXON	DEXTER CLASSIFICATION
	DSKIP4	DESKTOP IN CONTRACT
	GYRO	GREEK CLASSIFICATION
	JOANS	JOANS COMPUTER NETWORK
	SERUCE	SERVICE CLASSIFICATION IS FOR YOU
	TECHNO	TECHNO CLASSIFICATION
	UNCLAS	UNCLASSIFIED MATERIAL

Figure 5-13 Classification Pop-Up Window

Data Junction is entered as the *Long Title*. Since the *Long Title* can be up to 130 characters in length it is not possible to view the entire title on one line. Therefore, a feature has been added that allows the user to press the *TAB* key before or during *Long*

Title entry and view the entire title at once. A dialog box appears in the center of the screen with the *Long Title* wrap on two lines.

The entry of the next nine data fields requires little more than typing the data in and pressing *Return*. However, once the *Custodian* data field is reached, another pop-up window appears with the current custodians in the system (Figure 5-13). The user can either touch-type the first few letters of the *Cust ID* desired or scroll through the list with the arrow keys until the custodian is found who is responsible for the current document.

The screenshot shows a terminal window titled "Add Document". On the left, there are several data fields: "Barcode: 1301", "Custodian ID: U-28094", "Long Title: DATA JUNCTION", "Originating Agency: TOOLS AND TECHNIQUES", "Location: SHELF #3", and "Barcode: N/A". A central pop-up window titled "Cust ID | Rank | Last Name" displays a list of custodians: 1554 LT LOUIGIE, ELKERN94 LT ELKERN, J8844 LT SUANSON, JIM98 LT JENKINS, STEIN94 CIU STEINBECK, VENADR CIU ANDREWS, and WILSON LT WILSON. To the right of the list is a vertical scrollbar. Below the pop-up window, there are fields for "Status: NO", "Custodian: ACTIVE", and "Barcode: N/A". At the bottom of the screen, there are buttons for "New", "Keywords", and "Exit".

Cust ID	Rank	Last Name
1554	LT	LOUIGIE
ELKERN94	LT	ELKERN
J8844	LT	SUANSON
JIM98	LT	JENKINS
STEIN94	CIU	STEINBECK
VENADR	CIU	ANDREWS
WILSON	LT	WILSON

Figure 5-13 Custodian Pop-Up Window

Of the last three data fields to enter, the *Abstract* is the most important. The *Abstract* as discussed previously is used to assist user in finding what document they want. The *Abstract* provides a brief description about the document that can be viewed instantly, where as to gain a similar understanding of a document without the abstract, the user might have to page through the contents after requesting it from the librarian. The *Abstract* data field expands as does the *Long Title* by pressing the *TAB* key. The librarian

can enter a free form description of the document as lengthy as he/she wants. The *Long Title* is automatically inserted into the *Abstract* data field allowing the librarian to add any additional remarks (Figure 5-14). Once the *Abstract* is complete, pressing the *TAB* key saves its contents.

The screenshot shows a terminal window titled "Add Document". At the top, it displays document metadata: "Doc No: 1391", "Control No: U 28094", "Sheet Title: DAT-JUNC-94", and "Classification: UNCLAS". Below this, there is a section for "Long Title" with the text "DATA JUNCTION:". The main body of the screen is a large text entry field containing the following text: "Data Junction uses its 'Hub & Spoke' design to transfer data among a wide variety of software file types. The hub program is called the 'Junction.' The spokes that feed into and out of the hub are specialized conversion programs. The spokes are used to convert external file types into Data Junction files, and to convert Data Junction files into external file types." Below the text entry field, there are several status indicators: "Status: N/A", "Abstract: MAXIMUM", and "Keywords: ". At the bottom of the screen, there are four buttons: "New", "Keywords", and "Exit".

Figure 5-14 Add Document Abstract Feature

The next step after entering the document's characteristic data is to either add a completely *New* document, associated document *Keywords*, or *Exit* the *Add Document* feature. If *Keywords* is selected, the screen changes to the *Keyword Entry* screen. The layout of this screen is very similar to that of the Boolean search screen (Figure 5-1). The *Keywords* dialog box incorporates the touch type feature and allows users to view the current *Keywords* in the system. After the user types a few letters of a new keyword, either a match is found or the user can complete the entry and add it to the list of keywords. The *Keywords* already assigned to the current document appear on screen as well. This feature assists in standardizing keywords by allowing the librarians to view the

current keywords in the system and assign them to other documents prior to creating new keywords which make be completely unique to the current document. The example of "USSR" applies in this situation. If the librarian sees "USSR" as a keyword already in the system, he/she will be more likely to select it than to enter, "U.S.S.R."

If the user chooses *New* document a dialog box appears on screen asking the user if he/she wants to save the current document (*Yes* or *No*), or modify the current document. If the user responds with *Yes* and *Keywords* do not exist for the current document, a dialog box appears prompting the user to enter *Keywords*. Selecting *No* causes another dialog box to appear (Figure 5-15). This check-box options allow the user to either start a completely new document or to enter a series of documents based on the one previously entered. In larger libraries at training commands this feature is essential since they might receive 20 copies of the same document. It is senseless to require librarians to enter each of the 20 documents, especially when the only difference is their barcodes. When *New Barcode (Previous Document on Screen)* is selected, the user can specify the number of copies to enter. If the barcodes are sequential LDS adds the documents immediately. However, if the barcodes are not sequential, the librarian is prompted to enter each individual barcode before the documents can be entered into the system.

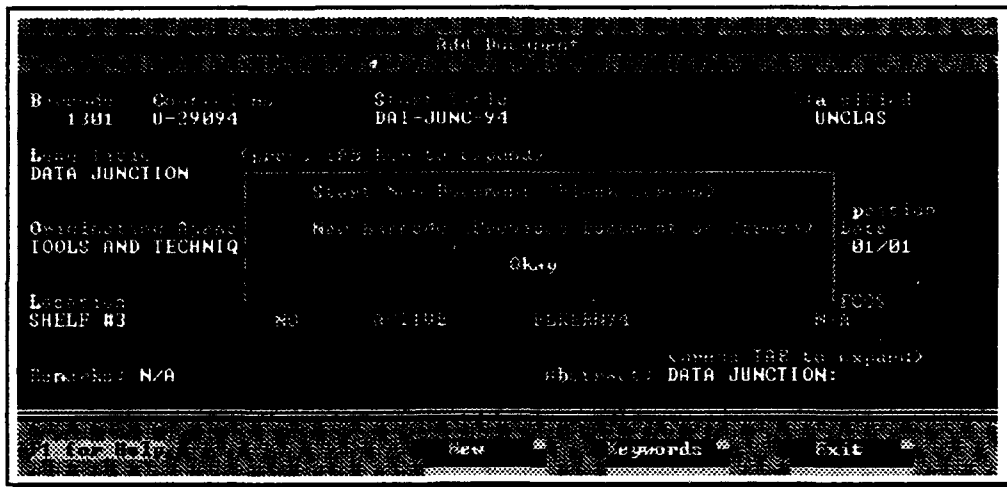


Figure 5-15 New Document Choices

If the user selects *Exit*, and the current document has not been saved, the same options dialog box appears prompting the user to either save, modify or no save the current document. After the user makes his/her choice the *Add Document* feature is left and the system returns to the *Main Menu*.

F. ON-LINE HELP

The new version of LDS is designed to provide users with access to context sensitive help from anywhere in the program by pressing the *F1* key. This will enhance users productivity from the moment they begin using LDS and continue to aid them as the venture into unfamiliar or less frequently used features. Although a comprehensive user's guide for LDS will be available, implementing the *On-line Help* should minimize the need for user's to access the manual.

The *Help* example is taken from the Boolean search feature where search composition can be somewhat difficult without adequate assistance. After selecting the

first keyword, the *Boolean Options* dialog box appears. For a new user this can be somewhat confusing, until the *F1* key is pressed and the help screen appears (Figure 5-16). The user can scroll-up/down reading as much or little help text as necessary to explain the current process. If further help is available on the current procedure or a related process, the *F1:NextHelp* message will appear at the bottom-right of the help screen. By pressing the *F1* key again, a pull-down menu appears at the top-left corner of the help screen. This menu is effectively a hypertext link to related processes that may further assist the user.

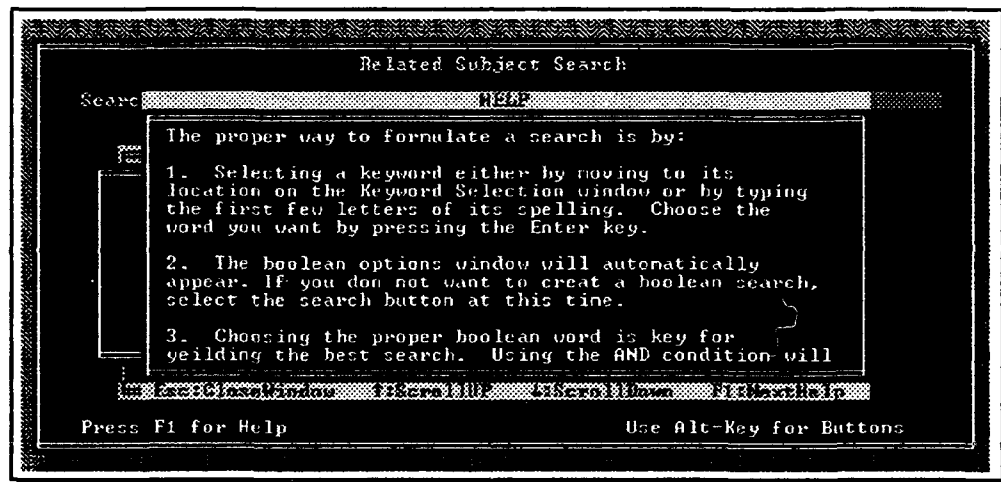


Figure 5-16 Sample Search Help Screen

VI. CONCLUSION

This thesis presents research that involved conducting a survey of DoN classified libraries to determine their automated systems needs. As a result, an extended automated classified document system (ACDS), called the Library Document System (LDS), was developed. LDS incorporates the latest in user interface technologies and database design methods to establish a stable system for usage and further development by DoN Libraries.

A. RESEARCH SUMMARY

The determination that a standardized ACDS does not exist in the Navy provided the impetus for developing and implementing LDS. Although other systems were discovered during research, they proved to be either too costly in terms of new equipment requirements and system size, or out-of-date with limited functionality and poor user interfaces.

The methods patrons use to gain access to classified libraries were identified. Although these methods are relatively effective, they often waste the patron's and librarian's time. This spurred the development of a sophisticated Boolean search mechanism for LDS. The amount of information presented via the Boolean search to users on-line would significantly reduce the need to page through documents in order to assess their content.

Information acquired from libraries outside DoN proved useful in assessing the availability and functionality of commercial ACDSs. It became evident that purchasing a commercially developed system is no guarantee of its effectiveness.

The ability to incorporate features typically associated with commercial systems was a major factor in the development of LDS. These features include: the Boolean search, user definable classifications, audit trails, and mousable user interfaces. Through the implementation of these features, LDS's future as a reliable ACDS has been secured.

B. APPLICATION

Although LDS was developed in the Navy, its application is DoD-wide for libraries that manage classified as well as unclassified documents. Because of its flexible configuration, a library in the Air Force could use LDS equally as well as a classified library in the Navy.

Database researchers may also find LDS interesting, as it is effectively a case study into how to upgrade an existing database system. The evolution of LDS from a database with seven tables to 17 was a tedious endeavor. This modification required meticulous attention to the relational database integrity and traditional database design methodologies.

The Boolean search algorithm might also be of interest to database programmers who are involved in creating a search mechanism for their systems. The new LDS search provides *AND* and *OR* search capabilities which can be used as a model for other systems.

C. FUTURE RESEARCH

Database evolution is an ongoing process both in theoretical and practical terms. This thesis represents the practical evolution of a database system, LDS. The need to enhance LDS with new database doctrines remains as long as users employ LDS as a tool. An example of applying a new database doctrine to LDS would be to transform LDS completely from a relational database system to the object-oriented database paradigm.

User interface design is another area which draws a great deal of attention. In today's PC market, if programs are not introduced in a Microsoft Windows™ graphic user interface format, their chances for success are seriously hampered. However, LDS and its market are very different, the results from the surveys indicated that the majority of potential users run MS-DOS 5.0 as their primary operating system. Therefore, the LDS upgrade was written to operate effectively, in MS-DOS with the option of running it under Windows. This solution bridges the gap between DOS and Windows users for the short-term. In the coming years, however, as DoN transitions to Windows based platforms, LDS will need to do the same.

Secure environments like classified libraries are becoming more and more interconnected with outside entities. Since LDS was designed to operate in a secure space, it does not have the multi-level secure (MLS) system built-in that would screen user clearance levels and provide appropriate access. A basic screening process was implemented that enables librarians to verify patron's access levels based on the patron's

profile stored in the *PATRON.DBF* table. However, this implementation is not required and only assists in authorizing access, it does not deny user access as in a MLS system. The area of creating MLS systems is one in which LDS would make a good candidate, especially if its network capability is exploited to provide access to network users who may not have the proper security clearances.

Further automation of LDS's search and check-in/out capability are areas that may be of interest. Reducing the amount of workload placed on librarians is the driving force behind LDS, however further reduction can be gained if the above tasks are automated. As the DoD goes through a reduction in work-force, providing patrons the ability to conduct their own searches would reduce the librarian's involvement until a document is requested. Although somewhat more difficult to anticipate, the automation of the check-in/out of routine material would be beneficial to librarians as well. A thorough analysis LDS and its time intensive areas would be another direction in locating the areas for further automation.

APPENDIX A: LDS SURVEY



LDS

Library Document System

LDS Configuration Survey

Command Name: _____

Address: _____

Point of Contact: _____ Phone: (DSN) _____

Alternate POC: _____ (Comm.) _____

Describe your library:

1. How many publications do you maintain in your collection? _____ (approximately)
2. How many patrons do you serve daily? _____
3. Do patrons have the ability to access your library automation system? ☐ YES ☐ NO

Describe the computers used for controlling your publications: (circle all that apply)

4. Computer Hardware used? (Intel 286/386/486) Brand name: _____
5. Operating System used? (DOS/Windows/OS-2/Novell/Unix) Version: _____
6. Are you on a network of Personal Computers? ☐ YES ☐ NO
If Yes, what Network are you using? (Novell/Lantastic/3Com/Other: _____)
7. What type of printer do you have? (Dot Matrix/Laser/Daisy Wheel/Other: _____)
Model and Brand name of printer: _____
8. Do you use a mouse frequently? ☐ YES ☐ NO

Describe your Library Automation Software:

9. Which library automation software are you currently using: (check one)
☐ CDC System ☐ dBase IV/III+ ☐ In-house system ☐ None
☐ Other (please specify) _____
10. Are you pleased with the Library Automation System you are using? ☐ YES ☐ NO
11. Have you had problems with your Library Automation System? (check all that apply)
☐ Not enough features
☐ Loses data (sometimes/frequently)
☐ Not user friendly or software is difficult to learn
☐ Worthington TriCoder (T-50) does not work as advertised
☐ Subcustody feature inadequate

UNCLASSIFIED

UNCLASSIFIED - Please do not answer questions that would violate security standards

☐ Other (explain): _____

12. If you checked #11 - "Not enough features", what additional features would you like?

☐ Boolean Search Capability, i.e., search by "Ships *and* Subs *or* Aircraft not anti-"

☐ Automatic Backup Option after hours

☐ Patron Access to Search of Library Collection

☐ Software would be mouse driven

☐ Other features (explain): _____

Library Document System software upgrade:

13. Would you be interested in upgrading from your current Library Automation System to a system that incorporates many of the features described above and has been in operation for nearly five years? ☐ YES ☐ NO

14. If your current system's data was converted to the proposed LDS system at a remote secure site, would you be more likely to upgrade? ☐ YES ☐ NO

15. Would a 3 letter classification field size be sufficient to handle the variety of classifications for publications at your command? ☐ YES ☐ NO

If no specify size: _____

16. Would you need to maintain more than 20 different levels of classified material on your library automation system? ☐ YES ☐ NO

If yes specify size rounding up to the nearest 10: _____

17. Would a Short Title size of 30 be sufficient to handle your publications? ☐ YES ☐ NO

If no specify size: _____

18. Do you subcustody and transfer publications to other personnel or commands often? ☐ YES ☐ NO

19. Do you need password protection for deleting publications or other areas? ☐ YES ☐ NO
If yes please specify other areas: _____

20. Are there any other specific concerns you have about upgrading to LDS?

(Attach Additional Sheets If Necessary)

UNCLASSIFIED

SURVEY RESULTS

The purpose of this document is to present the survey results from CDCS's users. The insight acquired from these results greatly assisted in planning, developing and implementing the upgraded LDS.

The survey questions were as follows:

1. *How many publications do you maintain in your collection? _____*
(approximately)

This answer varied from: (overall average: 1168)

2	0,	1	100,	1	225,	2	250,	1	255,	1	300,
1	500,	1	800,	1	1000,	1	1075,	1	1200,	3	2000,
1	2400,	1	2500,	1	3000.						

2. *How many patrons do you serve daily? Average 44*
20, 20, 5, 12, 12, 3, 160, 20, 11, 35, 10, 5, 40, 60, 5, 300, 15 = 733

3. *Do patrons have the ability to access your library automation system?*
ALL RESPONDED NO!!!

4. *Computer Hardware used? (Intel 286/386/486)*

286	1
386	13
486	3
VAX	1

5. *Operating System used? (DOS/Windows/OS-2/Novell/Unix)*

DOS 3.1:	1
DOS 5.0:	9
DOS 6.0:	2
DOS 6.2:	1
Windows:	2
VMS:	1

6. *Are you on a network of Personal Computers?*

Novell	2
Latin	1
Banyan	1

7. *What type of printer do you have? (Dot Matrix/Laser/Daisy Wheel/Other: _____) Model and Brand name of printer:*

Multiple types: 1

Laser:

North Atlantic II-T/III-T	6
Hewlett Packard Laser II-T	5
HP Laser Jet III/IIIP	3
UNISYS	1,
Mitek Model 110-T	1,

8. *Do you use a mouse frequently?*

Yes: 7

9. *Which library automation software are you currently using: (check one)*

- CDC System	15
- dBase IV/III+	1
- In-house system	1
- None:	1
- Other	WP51, Oracle,

10. *Are you pleased with the Library Automation System you are using?*

No:	9
Yes:	8

11. Have you had problems with your Library Automation System? (check all that apply)

- Not enough features 10
- Loses data (sometimes/frequently) 7
- Not user friendly or software is difficult to learn 10
- Worthington TriCoder (T-50) does not work as advertised 6
- Subcustody feature inadequate 4

-Other (explain):

- Unable to modify certain fields after initial data entry
- Fields not long enough to enter information to ease retrieval
- Labels should print sequentially w/o manually typing each label into the computer
- Use alternate GENSER registered system (not workable)???
- Doesn't produce appropriate forms

12. If you checked #11 - "Not enough features", what additional features would you like?

- Boolean Search Capability 10
- Automatic Backup Option after hours 7
- Patron Access to Search of Library Collection 8
- Software would be mouse driven 7

• Other features (explain):

- Universal barcode software compatible
- Search by subject/sort by category or country or subjects that are similar
- Inventory report is hard to read
- Need to be able to sort documents that pertain to certain subjects
- If a field was available to enter a category or sort by when requested or printing reports. when
- Field that states last inventory / date / by
- Would like the capacity to generate a destruction log
- Scanner should be easier to use within program
- Backup feature is very important with easy data recovery
- Should provide technical information to aid in trouble shooting on site, including file structure and the ability to recover data if the system crashes
- Automation of forms and use barcode technology?

13. *Would you be interested in upgrading from your current Library Automation System to a system that incorporates many of the features described above and has been in operation for nearly five years?*

- YES 19
- NO 1

14. *If your current system's data was converted to the proposed LDS system at a remote secure site, would you be more likely to upgrade?*

- YES 11
- NO 5

15. *Would a 3 letter classification field size be sufficient to handle the variety of classifications for publications at your command?*

- YES 15
- NO 5

-If no specify size: 5 -- The new size will be FIVE characters long.

16. *Would you need to maintain more than 20 different levels of classified material on your library automation system?*

- YES 2
- NO 18

17. *Would a Short Title size of 30 be sufficient to handle your publications?*

- YES 17
- NO 2

18. *Do you subcustody and transfer publications to other personnel or commands often?*

- YES 16
- NO 5

19. *Do you need password protection for deleting publications or other areas?*

- YES 8
- NO 10

- Additional Input:

- Various documents require "need to know" access
- Custody transfer
- Not necessary, but it should be made difficult to accidentally delete any pub, also undelete feature
- Various compartments
- House-keeping function

20. *Are there any other specific concerns you have about upgrading to LDS?*

- As long as we can be assured that Documents won't be lost, it should be upgraded and a Navy wide program that ADP Personnel are familiar with
- Support - availability for long-term support
- Maintainability - flexibility - expandability - compatibility with existing configuration
- Programmed access controls (discretionary access control)
- Can not be used in a GENSER registered system unless changed from present configuration
- Would like to use LDS before transfer
- Difficulty
- Prefer to perform data conversion on-site
- Will be upgrading our PC, purchasing a Gateway 2000, model 486-33

APPENDIX B: LDS DATABASE SCHEMA

1. Table for the DOCUMENT.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	BARCODE	Numeric	7	d_barcode
2		CONTROL_NO	Character	15	d_cntrl
3		LONGTITLE	Character	130	d_long
4		SHORTTITLE	Character	30	d_short
5		ORIGINATOR	Character	20	d_origin
6		DATEOF PUB	Date	8	
7		DATERECVD	Date	8	
8		COPYNUMB	Numeric	3	
9		COPYAVAIL	Numeric	3	
10		LOCATION	Character	15	
11		COMMAND_ID	Logical	1	
12	FOREIGN KEY	CLASS_NUMB	Numeric	3	d_clasmb
13	Active, Lost, Destroyed	STATUS	Character	1	
14		CUSTODIAN	Character	8	d_custs
15		DISPO_DATE	Date	8	
16		DEFCOS	Character	10	
17		NOTES	Character	30	
18		ABSTRACT	Memo	10	

2. Table for the SUBJECTS.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	SUBJ_NUMB	Numeric	2	
2		SHORTITLE	Character	30	s_short
3	FOREIGN KEY	KEY_NUMBER	Numeric	5	

- Index s_keysht = KeyNumber+Shortitle
- Index s_nmbsht = Subj_Numb+Shortitle

3. Table for the KEYWORDS.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	KEY_NUMBER	Numeric	5	k_keynmb
2		KEYWORD	Character	25	k_keywrđ

4. Table for the CUSTDIAN.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1		CSSN	Numeric	9	
2		CLASTNAME	Character	25	cu_lname
3		CFIRSTNAME	Character	15	
4	PRIMARY KEY	CUST_ID	Character	8	cu_id
5		CRATE_RANK	Character	6	
6		CLOCATION	Character	15	
8		CPHONE	Numeric	12	
8		CDATEINPUT	Date	8	
9	Primary or Sub.	CTYPE	Character	1	
10		CACCESS	Boolean	1	

5. Table for the PASSWORD.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	CUST_ID	Character	8	pw_cstid
2		PASSWORD	Character	15	

- This file will be hidden in the LDS directory to assist in keeping the passwords relatively secure.

6. Table for the ACCESS.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	ACCESS_SSN	Numeric	9	ac_ssn
2		ACC_LEVELS	Character	5	

7. Table for the CHKINOUT.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	SSN	Numeric	9	ck_ssn
2	FOREIGN KEY	BARCODE	Numeric	6	ck_bar
3	Type of check-out	CKSTATUS	Character	7	
4		DATEOUT	Date	8	ck_date
5		QUANTITY	Numeric	3	
6	Used with special barcodes only, classification determined at checkout	CLASSIFIC	Character	1	
7	Subcustody? T/F	CHK_OUT	Logical	1	
8	Custodian ID	CUSTODIAN	Character	8	

8. Table for the SPECIAL.DBF:

	Field Attributes	Field Name	Type	Width	Index File
	Based Rec. Numb.	RECNO()			sp_recno
1	PRIMARY KEY	BARCODE	Numeric	6	sp_bar
2	FOREIGN KEY	SHORTITLE	Character	20	

9. Table for the DOC_DEL.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	BARCODE	Numeric	7	dl_bar
2		CONTROL_NO	Character	15	ld_cntrl
3		LONGTITLE	Character	130	
4		SHORTITLE	Character	30	
5		ORIGINATOR	Character	20	
6		DATEOF PUB	Date	8	
7		DATERECVD	Date	8	
8		COPYNUMB	Numeric	3	
9		COPYAVAIL	Numeric	3	
10		LOCATION	Character	15	
11		COMMAND_ID	Logical	1	
12		CLASS_TYPE	Character	4	
13		STATUS	Character	1	
14		CUSTODIAN	Character	8	
15		DISPO_DATE	Date	8	
16		DEFCOS	Character	10	
17		NOTES	Character	30	
18	FOREIGN KEY	REPORT_NUM	Character	12	

- Effectively the same table as **DOCUMENT.DBF**, however the report number is added and is keyed to the **REPORTS.DBF** via the **REPORT_NUM**. See **REPORTS.DBF** for detail of it's stored information. The **ABSTRACT** field has been omitted as well for space conservation reasons.

10. Table for the REPORTS.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	REPORT_NUM	Character	12	r_number
2		RCUSTODIAN	Character	8	r_cust
3		RDATE	Date	8	
4		RTIME	Numeric	5	

11. Table for the COMMAND.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	COMMAND_ID	Character	6	
2	provide sample	COMD_NAME	Character	55	
3		DESTROY_YR	Numeric	1	
4		AUTHORIZED	Logical	1	
5		AUTH_OFF	Character	40	
6		DESTR_OFF	Character	40	
7		WITNESS_OF	Character	15	
8	A: or B: drive	DISK_BKUPS	Character	1	
9		PRINTER	Numeric	2	
10	Do you want to enter users at check-out time?	ENTER_USER	Logical	1	
11	Delete inactive users after X yrs	PURGE_USER	Numeric	2	
12	special barcode on/off switch	SPECIAL	Logical	1	
13	Title class for all menus on system	TITLE_CLAS	Character	50	
14	Set security on/off	SECURITY	Logical	1	

12. Table for the HISTORY.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	HBARCODE	Numeric	7	h_bar
2	S=system, D=delete	HTRANS_ACT	Character	1	
3		HDATE	Date	8	
4		HTIME	Numeric	5	
5	FOREIGN KEY	HCUSTODIAN	Character	8	h_cust

13. Table for the INVENTORY.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	BARCODE	Numeric	7	inv_bar
2		CLASS_NUMB	Numeric	3	
3		LOCATION	Character	15	
4		STATUS	Character	1	

14. Table for the TEMPVENT.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	BARCODE	Numeric	7	

15. Table for the INV_SESN.DBF:

	Field Description	Field Name	Type	Width	Index File
1	PRIMARY KEY	CUST_ID	Character	8	ses_cust
2	N-normal, C - classified, L - location.	TYPE	Character	1	
3		DESCRIBE	Character	15	
4	Start recno()	START	Numeric	7	
5	End recno()	STOP	Numeric	7	

16. Table for the PATRONS.DBF:

	Field Attributes	Field Name	Type	Width	Index File
1	PRIMARY KEY	PSSN	Numeric	9	p_ssn
2		PLASTNAME	Character	25	p_lname
3		PFIRSTNAM	Character	15	
4		PRATE_RANK	Character	6	
5		PLOCATION	Character	15	
6		PPHONE	Numeric	12	
7		PDATEINPUT	Date	8	
8		PACCESS	Logical	1	
9		PEXPIRES	Date	8	

17. Table for the CLASSES.DBF:

	Field Description	Field Name	Type	Width	Index File
1	PRIMARY KEY	CLASS_NUMB	Numeric	3	cl_num
1	SECOND KEY	CLASS_ABRV	Character	6	cl_abrv
2		CLASS_DEF	Character	40	

LIST OF REFERENCES

1. COMPUTER ASSOCIATES: "CA-Clipper", Computer Associates International, Inc., 1989-1993.
2. ASHTON-TATE: "dBASE III PLUS", Ashton-Tate, Inc., 1984-1986.
3. ELMASRI, R. & NAVATHE, S.: "Fundamentals of Database Management", The Benjamin/Cummings Publishing Company, Inc., 1989.
4. STILAS: "Scientific and Technical Information Library Automation System", Sirsi Corporation, 1993.
5. GALAXY: "Galaxy Integrated Library System", Gaylord Information Systems, Inc., 1994.
6. YOURDON, E.: "Modern Structured Analysis", Prentice-Hall, Inc., 1989.
7. MAXCESS: "The Maxcess Library System", MAXCESS Library Systems, Inc., 1994.
8. CDCS: "Classified Document Control System", Fuentez Systems Concepts, Inc., 1991-1993.
9. MAILROOM: "The Mailroom Inventory System", Sverdrup Technology, Inc., 1989.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, California 93943	2
3. Department Chairman, Code CS Computer Science Department Naval Postgraduate School Monterey, California 93943	6
4. Professor Timothy Shimeall, Code CS/sm Computer Science Department Naval Postgraduate School Monterey, California 93943	6
5. LT Kenneth F. Elkern, Jr. Navy and Marine Corps Intelligence Training Center Building #420 Dam Neck Virginia Beach, Virginia 23461	2
6. Mr. Jim Bouzios, Code 0624 Building 2434 NETPMSA Pensacola, Florida, 32509	2
7. Mrs. Betty Snellenberg Navy and Marine Corps Intelligence Training Center Building #420 Dam Neck Virginia Beach, Virginia 23461	2
8. Mr. Kelvin Langley, Code 5K Office of Naval Intelligence 4301 Suitland Road Washington, D.C. 20395	1

9. Ms. Terry Smith, Code G54
Commander Naval Security Group
3801 Nebraska Avenue
Northwest, D.C. 20393

1