

Received w/Ltr Dated 2/26/90

LICENSING SUPPORT SYSTEM PROTOTYPE TEST REPORT

SCIENCE APPLICATIONS INTERNATIONAL CORPORATION

February 16, 1990



9003010443 900226
PDR WASTE PDC
WM-1

403

TABLE OF CONTENTS

	<u>Page</u>
1.0 Introduction	1-1
1.1 Purpose and Scope.	1-1
1.2 Background	1-2
1.2.1 Assumptions.	1-3
1.2.2 Prototype Test-bed Configurations.	1-4
2.0 Data Capture Lessons	2-1
2.1 Initial Processing	2-1
2.2.1 Completeness	2-2
2.1.2 Legibility	2-2
2.1.3 Unitization.	2-3
2.1.4 Accession Numbers.	2-3
2.1.5 Page Numbering	2-4
2.1.6 Duplicate Check	2-5
2.1.7 Initial Processing Conclusions and Recommendations.	2-5
2.2 Cataloging	2-7
2.2.1 General Observations and Recommendations	2-7
2.2.2 Title.	2-7
2.2.3 Document Type and Detailed Document Type	2-8
2.2.4 Document Date.	2-9
2.2.5 Document/Report Number	2-9
2.2.6 Author Organization, Recipient Organization.	2-10
2.2.7 Location	2-10
2.2.8 Publication Data	2-11
2.2.9 Descriptors.	2-11
2.2.10 Abstract	2-12
2.2.11 Document Characterization.	2-13
2.3 Image and ASCII Preparation.	2-14
2.3.1 Background	2-14
2.3.2 Prototype Process and Data Collection.	2-15
2.3.3 Analyses and Observations.	2-16
2.3.3.1 Quality of Source Documents.	2-16
2.3.3.2 Accuracy	2-17
2.3.3.3 What are LSS Documents Like?	2-18
2.3.3.4 Formatting Rules and Substitution Codes.	2-19
2.3.3.5 Image Scanning	2-25
2.3.3.6 Preprocessing.	2-27
2.3.3.7 Editing.	2-27
2.3.3.8 Microform and Word Processing.	2-30
2.3.4 Summary of Lessons Learned	2-30
3.0 Data Preparation and Load.	3-1
3.1 Data Preparation Process	3-1
3.2 Data Base Load	3-2
3.3 Observations and Conclusions	3-3

TABLE OF CONTENTS

(Continued)

	<u>Page</u>
4.0 Analysis of User Tests.	4-1
4.1 User Test Design.	4-1
4.2 Analysis Results.	4-4
4.2.1 Header.	4-4
4.2.1.1 Header Fields	4-5
4.2.1.2 Descriptors	4-8
4.2.1.3 Special Tools	4-9
4.2.1.4 Search Strategies and Search Aids	4-9
4.2.2 Full Text	4-12
4.2.3 Searching, Retrieval, and Post-Processing of Results.	4-14
4.2.3.1 Special Tools	4-14
4.2.3.2 Strategies.	4-17
4.2.3.3 Conclusions and Recommendations . .	4-18
4.2.4 Images.	4-18
4.2.4.1 Usefulness of Bit-Mapped Images in the LSS.	4-19
4.2.4.2 Display Functionality Requirements.	4-20
4.2.4.3 Image Conclusion and Recommendations	4-21
4.2.5 General	4-22
4.2.5.1 Partitions.	4-22
4.2.5.2 General Display Issues.	4-24
4.2.5.3 System Resources.	4-25
4.2.5.4 Need for Printed Copy	4-26
4.2.5.5 Conclusions and Recommendations . .	4-27
4.2.6 User Profile.	4-27
5.0 Capture System Procedures Development Input	5-1
5.1 Initial Processing.	5-1
5.2 Cataloging.	5-1
5.3 ASCII Conversion and Image Preparation.	5-2
6.0 Search and Image System Design Requirements Input	6-1
6.1 Search System	6-1
6.1.1 Query and Retrieval	6-1
6.1.2 Display	6-3
6.1.3 User Interface.	6-4
6.1.4 Resource Requirements	6-5
6.2 Image System.	6-5
7.0 References.	7-1

Appendices (Bound Separately)

LIST OF TABLES AND FIGURES

	<u>Page</u>
Figure 2-1 Sample of Original.	2-20
Figure 2-2 Sample of ASCII Conversion.	2-21
Figure 2-3 Sample of Original Table.	2-24
Figure 2-4 Dictionary Size Effects	2-29
Figure 3-1 Load Times versus Load Size	3-5
Table 4-1 User Tests Summary Statistics.	4-2
Table 4-2 Field Use (Questionnaire).	4-6
Table 4-3 Field Use (Actual)	4-10
Table 4-4 Full Text Use.	4-12
Table 4-5 Field and Full Text Use.	4-13
Figure 4-1 Frequency of Image Requests	4-20

1.0 INTRODUCTION

1.1 Purpose and Scope

This Licensing Support System (LSS) Prototype Test Report discusses the lessons learned during the prototype development efforts, including document preparation, header cataloging, image generation, ASCII text conversion, data base loading, and the concluding prototype user testing. The report also presents a detailed analysis of the results obtained during the prototype user tests which were conducted from October 2, 1989 through October 13, 1989. During this two-week user test period, the search and retrieval strategies and behaviors of 44 users were studied using the LSS prototype systems.

There were eight groups of test users, with up to six test users in each group. Each user participated in a two-day testing session on the LSS prototype systems. The two-day testing session consisted of the following:

- (1) An Introduction and Orientation briefing,
- (2) A hands-on user training session,
- (3) Two 3 and 1/2 hour testing sessions,
- (4) Completion of a user test questionnaire, and
- (5) A debriefing session with each test user.

Each test user was assigned to one of the following usage categories, as defined in the LSS Preliminary Needs Analysis:

- (1) Technical and Engineering Usage,
- (2) Regulatory and Licensing Usage,
- (3) Management and Administrative Usage,
- (4) Public Information and General Public Usage, and
- (5) Intermediary Usage.

Each user was given questions, consistent with their assigned usage category, to answer during their test sessions, using the search and retrieval capabilities of the LSS prototype system. During each user's hands-on test session, an automated computer log of search and retrieval actions was generated, including queries entered, result sets retrieved, displays and formats requested, and corresponding timings for each action. The test results for the LSS prototype user tests were analyzed using the following five types of information collected during each user test session:

- (1) Automated computer logs of each user test session,
- (2) Answers to each question identified by the users,
- (3) Written comments and notes, including user comments, made during each user session recorded by an SAIC observer,

- (4) A written questionnaire filled out by each user at the end of his/her two day test session, and
- (5) Written results of the debriefing session for each test user at the end of his/her two day test session.

Section 2.0 of this report discusses the lessons learned during the data capture phase of the prototype data base development effort, including initial processing, header cataloging, ASCII test preparation, and bit-mapped image preparation. As discussed in this report, an additional 30,000 pages of representative LSS data will be processed with procedures which will mimic those defined in the Capture System Design Document. This effort will be instrumented and closely monitored to yield additional detailed data regarding the cost and efficiencies of the data capture process. Section 3.0 discusses the ASCII text and image preparation and data base loading efforts. Section 4.0 presents the analysis of the prototype users tests. It discusses the user test design and the analysis of the results obtained from the five types of information collected during the test sessions. Section 5.0 summarizes the lessons learned from the data capture and data base preparation of the prototype test data base and their effects on the procedures and operations of the final LSS Data Capture Systems. Section 6.0 summarizes the lessons learned from the prototype user tests in the areas of user interface, header and full-text search and retrieval operations, bit-mapped image retrieval and utilization, data base partitioning, and formatting and display of retrieval results. These lessons learned will provide requirement inputs for the design and development of the LSS Search and Image system. The four Appendices to this report provide the statistics generated from the data collected during the user test.

1.2 Background

The LSS is to be a computerized management information system which will capture, store, access, and present (output) all records, regulations, and technical information that is deemed relevant to obtaining the necessary licenses and permits for the siting, design, construction, operation, and closure of a geologic repository for the disposal of spent nuclear fuel and high-level nuclear waste as authorized by the Nuclear Waste Policy Act of 1982 and the Nuclear Waste Policy Amendments Act of 1987. The preliminary functions and requirements for the LSS have been documented in four reports prepared by the DOE Office of Civilian Radioactive Waste Management (OCRWM). They are:

- o Preliminary Needs Analysis (February 1988)
- o Preliminary Data Scope Analysis (March 1988)
- o Conceptual Design Analysis (May 1988)
- o Benefit-Cost Analysis (July 1988)

The Preliminary Needs Analysis and the Preliminary Data Scope Analysis presented system requirements upon which the Conceptual Design was based. The Benefit-Cost Analysis evaluated alternative architectures for the Conceptual Design.

Because no computerized document management system for licensing support of comparable size and complexity exists, several important design questions and issues arose during the LSS conceptual design analysis that need resolution before the system design can be completed and specifications for hardware and software can be developed. To address these LSS system design questions and issues, a LSS Prototype Test Bed was defined and documented in the Prototype Development Plan (January 1988). The Prototype Test Plan (September 1988) outlined the design issues the prototype test bed would address and described the data to be collected, procedures for the data collection, and the context in which the data would be used. The underlying rationale for the prototype test bed include:

- o Build an experience base for defining procedures for the capture of LSS document text, images, and cataloged header data, and
- o Provide LSS developers and users the opportunity to experiment with and to refine concepts of how people will use the LSS.

To support these underlying rationale, the prototype test bed was used for user tests to gain a deeper understanding of the following five broad questions relative to LSS:

- (1) How can data capture and preparation (cataloging, conversion, and loading) be made as cost effective as possible?
- (2) What search and retrieval strategies will users employ?
- (3) How will users use images?
- (4) What system resources are required?
- (5) What is LSS data like? (ie, document types, average number of pages and words per page, graphics content, etc.)

1.2.1 Assumptions

For the LSS Prototype development and testing, the following assumptions were made:

- (1) The prototype data base would consist of approximately 100,000 pages of relevant LSS documents that will be in the LSS records access

system during the early years of operation. A sufficient number of selected document bit-mapped images would be available on-line to support the prototype testing requirements for image retrieval and use.

- (2) OCRWM personnel and personnel from other potential licensing parties would be available to use the prototype to solve problems representative of those envisioned later in the licensing process.
- (3) The prototype would not be a small-scale replica of the operational LSS, but rather an instrumented test-bed designed to meet the prototype test plan objectives.
- (4) The LSS, and therefore the prototype, must accommodate both experienced and inexperienced users relative to familiarity with the LSS data base and use of document search systems. Therefore, user training, a brief description of the data base, and reference manuals would be provided.

1.2.2 Prototype Test-bed Configurations

The prototype test-bed consisted of two independent system configurations; (1) a remote system and (2) a local system. Both systems contained the same header and ASCII text data bases, but each used a different commercial-off-the-shelf (COTS) full text search and retrieval software package.

The remote system consisted of four workstations connected via telecommunication lines to a remote host which contained the header and ASCII text data bases and which executed the search and retrieval software package and customized user interface software. The user interface software permitted the user to build queries using pull down windows containing "picklists" of query terms from the document header and ASCII full text. The user interface also supported the typing in of queries using a native mode query language similar to that of the full text search and retrieval package. Two of the workstations provided only header and ASCII text search and retrieval capabilities, while the other two workstations provided both header and ASCII text search and retrieval, and image retrieval and display capabilities. The images were on optical disk systems directly connected to the two workstations.

Due to the projected size of the final LSS data base, it is envisioned that it will be composed of partitioned data bases to optimize the search and retrieval functions. For this reason, the remote prototype system's header and ASCII text data bases were partitioned into five data bases, based upon document type and date. This permitted the prototype testing and evaluation of user's search and retrieval strategies on a partitioned data base. The user interface software provided the ability to easily change partitions during a test session.

The local system consisted of a 386/25 PC containing the header and ASCII text data bases and executing the full text search and retrieval software package and a second PC running as a remote terminal connected to the 386/25 host. The local prototype system did not contain image retrieval and display capabilities. Also, the header and ASCII text data bases were not partitioned. The local system used the user interface provided by the COTS full text search and retrieval package and was not customized.

2.0 DATA CAPTURE LESSONS

The purpose of the LSS Capture System is to take the information provided by the submitters, define the proper LSS record, prepare complete headers for each LSS record, create an electronic image of each page of the LSS record and when appropriate, an ASCII text file of each page, and provide the results to the Search and Image System for loading into the LSS. The prototype, in addition to providing a test bed for user retrieval behavior, provided a valuable experience base for defining and testing procedures for capture of text, image, and header data. The lessons learned from this capture process are discussed in this section. Section 2.1 discusses initial processing of documents; Section 2.2, header cataloging; Section 2.3, Image and ASCII preparation.

One of the first steps in building the prototype was to determine what body of information would constitute the prototype data base. Two choices were apparent: (1) include representative examples of each type of document that would be submitted to the LSS, or (2) include a body of homogeneous information that might not represent every document type. The latter choice would provide a data base that would not only support gathering information needed for design issues, but also give test participants a narrow but integrated body of knowledge to search. Therefore, the Site Characterization Plan, its cited references, and its anticipated administrative record were selected as the body of information from which to acquire the approximately 100,000 pages that would make up the prototype data base. These documents were subsequently acquired from local governmental libraries and from the Department of Energy Office of Civilian Radioactive Waste Management (OCRWM) files, as well as from original sources. Photocopies were made where necessary for processing through the data capture system.

One of the functions of the prototype effort was to determine viable procedures for each step of the capture process. This was accomplished in all but the following areas: handling non-paper media, duplicate checking, and correction station operation. The other function was to answer two questions posed in the Prototype Test Plan: 1) how can data capture be made cost-effective, and 2) what are LSS documents like. By trying various configurations of the document flow through each process and by collecting data on different kinds of documents, the team could provide answers to these questions.

2.1 Initial Processing

As defined in the LSS Capture System Design Document, initial processing includes all steps needed to prepare a submittal unit for processing by scanning, ASCII conversion, and cataloging. This includes such steps as checking each document for completeness and legibility, unitization (determining whether a document needs to be subdivided), assigning an accession number to each unit, entering the basic information about the unit into the capture system, and performing a duplicate document check.

Armed with the knowledge that there are few "perfect" documents (i.e., those which contain perfectly legible text positioned squarely on 8-1/2" by 11" pages with no graphics, tables, or formulas), the team had a number of issues to resolve prior to the user test (e.g., how to handle oversized maps; where to place the accession number on a document). In addition, the process generated a number of issues which would be resolved based on input from the user test (e.g., do users expect to search scientific formulas in abstracts and document text; in cases where the document occupies only a portion of a page, do users expect to see the image of the entire page). These issues are discussed in the following subsections; conclusions and recommendations are presented in section 2.1.7.

2.1.1 Completeness

Documents could have pages or attachments missing; a field (Document Condition) in the header indicates this to the user. In addition, a document could have pages that could be eliminated. Such pages are the journal cover and table of contents for a single journal article or blank pages within the text (after the title page or table of contents). Especially in the case of a journal article preceded by cover or table of contents pages, processing these pages could be an unnecessary expense. The prototype data base contained 195 journal articles, which could have added between 195 and 390 pages to be optically stored (as well as the cost of any ASCII conversion). In addition, this could be a source of confusion to the user who, upon displaying the image and seeing the journal cover, might think the entire journal was entered into the data base.

2.1.2 Legibility

The processed documents had pages with varying degrees of legibility. Illegibility could be caused by poor photocopying of an original, photocopies of photocopies, or tightly-bound documents which did not allow for copying words that disappeared into the binding. Although a header field (Document Condition) captures the fact that a document is illegible, the degree of illegibility that must exist before a document is so designated is not easily determined. The initial rule was that a document had to be more than 50% illegible. This number rose to 75% during the course of the data base preparation when it became apparent that the number of illegible documents was greater than had been supposed. Although the submitter will be required to produce the "best available copy" for the LSS, documents in the millions of pages backlog may have illegible portions in the only available copy. A user searching in full-text might miss an important document because illegible words were not converted into ASCII and were, therefore, unsearchable. However, a user retrieving a document by a header search might be able to obtain desired information even though the edges of each page were not legible or lines on the bottom/top of the page were missing. There is also a point at which an illegible document is not at all useful to the user, as well as being too costly to process.

2.1.3 Unitization

Before any processing could begin, a determination had to be made as to what the basic unit of information to be processed would be, the premise being that each document would be a unit (and, therefore, one data base record). As long as the entire document was attributed to one author or a group of authors and treated one topic or a group of related topics, the treatment was apparent--the document was one processing unit. As soon as documents entered the system contained individually-authored parts or had attachments, rules had to be determined for treating these documents. It was evident that users would wish to search for both a collection of writings (such as a conference proceedings or an edited book) and also for the individually-authored papers or chapters contained in the collection. Therefore, a procedure was needed to allow the user to do so.

Unitization rules were devised to govern the process by which a document or group of documents would be subdivided into processing units, each unit becoming one data base record. Through unitization, one processing unit could be indicated for a collection of writings or a book and another unit for each separately-authored paper or chapter. Similarly, attachments to correspondence might be self-contained documents and, therefore, would become separate units.

It also became apparent that the same document might enter the system several times--as a single entity, attached to correspondence, as an appendix in a report, as part of a meeting package, and so on. Again, through unitization, this document would be treated as one processing unit, entered once into the system, and, by taking advantage of the pointer field in the header, would not need to be captured again. Thus, in addition to decreasing the cost of processing, proper unitization has the advantage of decreasing the number of duplicate hits from a user query while indicating to the user all occurrences of a particular document.

2.1.4 Accession Numbers

Various accession numbering schemes were proposed by team members, the most useful ones including numbering the units consecutively, incorporating the date into the number, using the submitter's accession number, or using a "smart" number (i.e., one that would reveal some information about the document). Regardless of the numbering method chosen, the number had to be unique. In addition, a decision was made to use the accession number as the file name for the image file(s) and for the ASCII file(s) containing the document text to facilitate linking the three pieces (header, text, and image) in the data base. This immediately limited the length of the accession number to fourteen characters (the number of characters supported by the UNIX operating system).

Since there was no way of ensuring that two submitters would not use the same number or that a submitter's number would not exceed the number of

characters available, it was decided that the submitter's number would not be used in this field but would be captured in another field (Submitter's Accession Number). An attempt was made at using a "smart" number--one that would indicate to the user that the document had other parts to it. This became very complex when documents appeared that were attachments to attachments and became impossible when a document appeared that was already in the data base. When it became apparent that document processing would begin before prototype hardware and software became available, a numbering scheme that would enable documents to be easily tracked through processing was the best choice. Consecutive numbering where every number had to be accounted for appeared to be the best scheme. In addition, this system enabled labels to be generated by a computer in a single run instead of generating them daily (which would be the case if a number incorporating the date was used). Labels containing a computer-generated consecutive number were printed and entered into a PC-based tracking system. As documents arrived for processing, the next available number was assigned. If a document was later found to be a duplicate, the number was retired with an appropriate notation.

Perusal of a large number of documents plus common industry practice dictated the placement of the accession number in the upper right-hand corner of the first page of the unit. This area appeared to be the least likely to contain text. Of course, there were instances in which this area did contain text. In these cases the number was placed as close to that corner as possible without obliterating any text. (This rule will also apply to LSS documents, since each page will contain a combination accession/page number.)

2.1.5 Page Numbering

Page numbering appeared to be a simple task until the question arose of what is a page. Pages that precipitated the question were instances where two pages of text were photocopied on one sheet of paper, a page that did not contain any text, and pages that exceeded 8-1/2 x 11 inches. Initially, a page was any sheet of paper 8-1/2 x 11 inches. Therefore, two pages of text on one sheet of paper would equal one page for processing. Similarly, blank pages would be counted and scanned.

In the case where two pages of text appear on one sheet of paper (generally in landscape orientation), a decision was made to cut the sheet in two and mount each page on separate 8-1/2 x 11 sheets of paper (in portrait orientation) for processing. This prevents garbling of the ASCII text during ASCII conversion and gives the user the normal portrait orientation of a page. It also provides two images, each of which can be viewed without scrolling.

Initially, pages greater than 8-1/2 x 11 inches were tiled (that is, the page was photocopied beginning at the lower left-hand corner of the page and using about a one-inch overlap). It soon became apparent that tiling pages greater than 17 x 22 inches would not be useful to a user without some means of

orientation as well as the relationship of the current image to the original page.

A decision was also made not to count or process blank pages. Pages that contained the sentence "This is a blank page" would be processed as a page, as well as pages that merely contained a page number. Blank pages could easily be eliminated from single-sided documents, but removing blank pages from duplex documents (documents that were printed on both sides of a page) was impossible. If these pages are to be eliminated at the scanning level, pages will have to be hand fed. If the document is photocopied as a single-sided document, the blank pages can be eliminated prior to scanning. Probably the simplest and most cost-effective solution is to process the blank pages only in those instances where the blank page cannot be pulled out because there is text on the reverse side.

2.1.6 Duplicate Check

Since document processing began before prototype hardware and software became available, document acquisition tracking, accession numbering, and cataloging were performed on several separate microcomputers. Therefore, no duplicate check system was available. Duplicates were removed as they became apparent to catalogers or to the QC (quality check) cataloger (strictly through memory). A duplicate check algorithm still needs to be defined, preferably in time for testing during the capture system simulation associated with simulation of Capture System Procedures. Given the number of duplicates (72 or 2.6%) encountered in the prototype, a good duplicate check system would alleviate user exasperation at retrieving duplicate documents from a search query, as well as eliminate the cost of processing such documents. (Despite the good memory of the catalogers, several test users pointed out duplicates in their retrieval sets.)

2.1.7 Initial Processing Conclusions and Recommendations

Although the initial processing phase of the prototype document capture operation did not simulate or mimic a capture station as defined in the LSS Capture System Design Document, certain conclusions and recommendations can be drawn from the prototype experience that could make data capture more cost-effective, as well as affect the procedures that are to be used.

The most costly area of the capture system is the scanning and ASCII conversion process. The cost-effectiveness of this process can be increased by decreasing the number of pages that need to be processed or by eliminating time-consuming document preparation. From the initial processing aspect of document capture, there are several ways of decreasing either the number of processed pages or document preparation time while at the same time increasing user satisfaction with the system.

One cost-effective measure is to eliminate extraneous pages from document capture. These pages are either blank sheets, journal cover or title pages, or duplicate documents. Unless the blank page is necessary to assure the user that he is not missing any pages (e.g., pages i and iii are included but there is no intervening page since it was blank), blank pages should be eliminated from capture whenever possible. Journal covers, title pages, or tables of contents are not needed for a single journal article since the pertinent information (journal title, volume, issue number, and date) generally appear on the first page of each article. These pages should, therefore, be eliminated. Similarly, no purpose is served by entering the same document more than once into the data base. Documents that appear in more than one context will have that context maintained through the LSS Pointer field without processing the document again. In addition to reducing processing time and costs, the user would have the advantage of reducing the number of hits (duplicates) resulting from a query.

Another way of increasing the cost-effectiveness of the system is to reduce document preparation time. Documents that are journal articles, book chapters, conference papers, or excerpts from a page can be time-consuming to process if there is non-relevant text (i.e., text from a previous or subsequent article, paper, or chapter) on the first or last pages of the articles, chapters, or papers or surrounding the excerpt on a page. Various procedures were tested during the prototype (blocking out the extraneous text, cutting extraneous text away from an excerpt and mounting it on a separate sheet of paper, etc.). Since users did not express a preference in this area, the recommendation is to image the page as it exists in the larger work and delete the extraneous text from the ASCII during ASCII conversion. This eliminates document preparation time (for masking irrelevant text or cutting pages to mounting the relevant text portions) without adding an equivalent amount of time to the ASCII conversion process.

Another cost-effective measure is to eliminate documents that are largely illegible. Documents that do not contain enough legible text to complete the duplicate check fields in the header should not be captured. In addition, pages that are largely illegible in a document that may contain many legible pages should be imaged only with an "unreadable text" notation in the ASCII. An attempt to rekey text that is not comprehensible does not serve the user if it leads to false hits.

Another recommendation for ease of processing is to treat landscape-oriented pages as if they were portrait-oriented for the purpose of placing the accession number on the document. This means that the accession number would be placed in the bottom right-hand corner perpendicular to the text in a page that is landscape-oriented. This eliminates the need for manual placement in a document that has pages of mixed orientation and should not have any effect on a user's perception of an image.

A final recommendation is to determine a duplicate check algorithm in time to test it during the capture system simulation. Although the capture system hardware and software will not be available for this effort, the prototype header

data base will be available, and the cataloging data base can be made searchable in order to test various algorithms for use in the capture system.

2.2 Cataloging

The cataloging process comprises entering information from each unit into the appropriate header fields and performing a quality check on the information entered. For the prototype, each cataloger had two tools in addition to the cataloging software: the LSS Prototype Cataloging Manual and the LSS Prototype Thesaurus. The LSS Prototype Thesaurus contained the terms available for entry in the descriptor field. The LSS Prototype Cataloging Manual was divided into two parts, one that explained how to use the software and the other that contained the rules for entering data into the fields. The cataloging software presented the cataloger with a screen of fields with space for entering data in each field. Fields were ordered so that the information taken from the cover or title page of the document were listed first. The accession number and title were the first two fields and remained in a box at the top of the screen as a reminder to the cataloger. If a field was a required entry, the cataloger could not bypass the field without the proper data entry. If a field entry was governed by a controlled vocabulary, the vocabulary appeared in a window from which the cataloger selected the appropriate entry or entries. This eliminated improper entries and misspellings of correct entries. Cataloging issues and recommendations are discussed below in the context of the specific field.

2.2.1 General Observations and Recommendations

In general, catalogers had some of the same observations as test users. They wanted a better interface, fewer keystrokes to complete entries, larger windows, function keys that had consistent operations across the cataloging operation, word processing features for text fields, and aids such as an online thesaurus with "explode" and "go to" capabilities, online help (application-specific pages from the manual), word counts from the document, and searchable ASCII for the unit being cataloged. In addition, consideration should be given to reordering the fields in the header, e.g., title, publication data, and abstract should appear consecutively. Windows should appear automatically when a cataloger selects a field, and, upon typing the first few letters of the desired entry, that entry should immediately be highlighted in the list without the cataloger using a "go to" key. This would speed up the cataloging process and take advantage of a cataloger's expertise on the system.

2.2.2 Title

For sorting purposes, the words "The" "A" and "An" were omitted as the first word in the title (unless they were an integral part of a phrase, such as "The Geysers"). These words were placed at the end of the title, an awkward

solution. The recommended solution is to enter the title as it appears and have the sort routine in the Search and Image ignore the article.

If the processing unit was a part of a larger work, the title field only contained the title of the unit; the title of the larger work appeared in the Publication Data field. It was thought that the title of the unit would not therefore, be confused with the title of the entire work. This was not the case, however. It is recommended that in cases where the entire document is attributable to a single author or group of authors, the title of the unit and the title of the document containing the unit both appear in the Title field. If the document contains separately authored units, the title of the encompassing document will remain in the Publication Data field. However, it is recommended that the user have the capability of searching both fields as a single unit or searching the two fields individually. This will allow the user who is searching for all parts of the Site Characterization Plan, for example, to find them by querying the title field. On the other hand, the user who is searching for documents containing the word "geophysical" in the title will not get every article from the Journal of Geophysical Research if the query is limited to just the Title field. In addition, the user who wants everything from a particular waste conference can search both fields as a single unit to retrieve all papers from the conference as well as the master record for the conference.

2.2.3 Document Type and Detailed Document Type

The controlled vocabulary for the Document Type field is the one currently in use by OCRWM in their records management system. Although definitions were given with each document type, catalogers were often unable to distinguish between some of the document types. For example, is a letter status report "Correspondence" or "Reports"; is the Site Characterization Plan a "Governing Document" or a "Report"; is a public hearing report a "Report" or "Legal Materials".

The Detailed Document Type field compounds the problem. Not only were the same problems present in distinguishing between choices at this level, but units were being processed that did not fit any detailed document type (e.g., units such as rulings, regulatory guides, Senate or House bills, Congressional conference reports, charters, etc.). It is recommended that the Document Type and Detailed Document Type fields be revised to contain entries that will not be confusing to the user. This is doubly important if the data base is to be partitioned by document type. If users do not understand what the different document types are, they cannot be expected to select the right partition for a search. Such revision will also benefit the catalogers who had similar difficulty in assigning the document/detailed document type to a document.

If the Document Type field remains a two-part field (Document Type and Detailed Document Type) the selection of one of the parts should be automatic. Either the document type is assigned automatically when the cataloger selects

a detailed document type, or, upon selection of a document type, the window for the detailed document type will contain only the appropriate choices for that type of document.

2.2.4 Document Date

The date field began as an eight-character field (yy/mm/dd) with the year preceding the month and day to allow range searching. Following existing OCRWM rules, the field required a valid month and day entry ("00" was not a valid entry). The OCRWM rule was to use yy/01/01 in the field if the date of the unit was a year only, and yy/mm/01 if the day was not given. Several results became apparent: (1) the date in the header would not match the date of the ASCII text or the image, possibly leading to confusion on the part of the user; (2) a user searching for a document published in January of a year would retrieve all documents that were published during that year that did not contain a month or day; (3) if a user sorted by date, documents issued in January would be intermingled with documents issued during the year that had a year only date. The decision was made to use "00" for an unknown portion of the date. In addition, since some documents in the system were issued during the 1800s and since the data base would eventually be covering documents issued in the next century, the field was increased by two characters and the format became cyy/mm/dd.

Many documents contain event dates as well as publication dates. For example, a document may be published in one year but report on an event (meeting, conference, hearing, etc.) that occurred the previous year. Addition of an "Event Date" field is recommended; this should be a repeating field.

2.2.5 Document/Report Number

In general, document numbers were entered into this field in the same format as they appeared on the document, using any marks of punctuation that the number contained. When it became apparent that there was not always consistency even within some of the government agencies, certain number series were standardized so that all numbers in a series had the same marks of punctuation within the number. This arrangement still presented problems to both the user and the cataloger searching for a particular number (see Section 4.2.1). Having a pick-list of all numbers available to users would solve the problem for novice or occasional users, but not for experienced users relying on a command or type-in mode. It is recommended that punctuation marks within report numbers be invisible to the search software (e.g., if USGS/OFR-84-1324 is entered, it would be "seen" as USGSOFR841324 by the software). (This same concept should be true of other areas containing hyphenated words, e.g. J-13 Borehole or well UE-25c #1.)

2.2.6 Author Organization, Recipient Organization

Names of organizations had to be standardized and be as brief as possible without losing their identity. Current OCRWM rules were generally adopted with a few minor exceptions. One of the rules for associations and government agencies was to use the common acronym. In some instances, however, two organizations share the same acronym (e.g., Nuclear Regulatory Commission and National Research Council, National Education Association and Nuclear Energy Agency). In all such cases except that of the Nuclear Regulatory Commission, the acronym was not used for either agency. User behavior shows that users are rarely aware of other possible meanings for their search terms (for example, a lawyer and a geologist will both expect "deposition" to relate only to their area of expertise). Therefore, users will expect to use the acronym with which they are familiar. This is true even when pick-lists are used. It is recommended in these instances that the acronym be kept for both organizations but be modified so that the user knows what he is selecting (e.g., NEA (Nucl. Energy Agcy) and NEA (Nat. Educ. Assn). A user should not receive a "not found" message upon entering the term "NEA" but should receive information enabling him to narrow his search to the right organization; nor should a cataloger be faced with a similar dilemma.

Some authors are affiliated with one organization but are writing as a representative of another organization. For example, a lawyer of a particular law firm might be writing a letter in his capacity as the representative of an Indian tribe, or an electric utility executive might be writing as a member of a particular electric power pool. Currently, these affiliations cannot be captured by the system. It is recommended that the author organization field should be a repeating field for each author to capture all such affiliations and still let the user query one field.

2.2.7 Location

The location field stores the location of the site of activity or the location to which the work described in the unit pertains, if pertinent. This field uses the LSS Thesaurus for terms in its controlled vocabulary. The idea in making this a separate field instead of including it as a descriptor was to ensure the assignment of such terms and to give users an easy mechanism for narrowing a search. The result, however, was to make the field too prominent, leading to the overuse of the field by catalogers. Additionally, since the field used a controlled vocabulary from the thesaurus, new terms were not added quickly enough to be useful, resulting in some place names appearing in the identifier field. This means that users had to search both location and identifier fields, (which caused confusion among some users). It is recommended that the location field be eliminated as a separate field and that the descriptor field be used to contain such information.

2.2.8 Publication Data

This field contains bibliographic information that is not covered in the fields above yet is important in identifying or citing the unit. Such information can include the title of the journal or book containing the unit, conference names and locations for proceedings, publishers, inclusive pages for a chapter or article, etc. The combination of the author, title, and publication data fields present the user with a standard bibliographic citation for use in a bibliography for an article being written, for review by an end-user, for acquisition of non-document items that have only a header in the data base, etc.

It became apparent during cataloging that confusion might exist over the use of the word "title" in this field. Users might expect the title of the journal or book to appear in the title field with the title of the article or chapter. Since this indeed turned out to be the case, it is recommended that this field be revised to reflect the proposed revision in the title field (see section 2.2.2).

2.2.9 Descriptors

The rules for assigning descriptors can have a great influence on both user retrieval and on the cost of cataloging units. If descriptors are assigned to represent only major concepts in the unit, cataloging cost is lower and user queries will result in retrieval sets with low precision (a greater number of non-relevant documents) but higher recall (more documents retrieved that contain some information). If descriptors are assigned at a very detailed level, cataloging cost will be higher and user queries will result in retrieval sets with high precision (most documents will be relevant) but low recall (the number of documents retrieved will be small compared to the number of documents containing some information on the subject). For the prototype test, it was decided to assign the descriptors at the detailed level and test the effect on users. This decision confirmed expectations that descriptor assignment would become very time-consuming (and, therefore, costly) and that catalogers would have to have much greater expertise in the various subject areas. Given the fact that full text is available to the user and that users will use full text to narrow their searches, it is recommended that descriptors be assigned that represent only the major concepts of the document. The intent is that users will retrieve a large set of potentially relevant records upon which full text searches can be conducted.

It was also quickly apparent that the thesaurus did not have enough definitions for terms (even in the eyes of the subject specialists) or enough cross-references (e.g., catalogers searching for "radioactive waste retrieval" or "waste retrieval" were not directed to the preferred term "retrieval of waste"). An effort is underway to remedy this.

Originally, it was believed that some means of grouping the terms in the thesaurus would assist both user and cataloger in finding descriptors. The ideal schema would be a faceted classification system. However, the number of disciplines represented in the thesaurus and the breadth of the terminology rendered this concept useless for our purpose. Instead, the terms themselves were reviewed in light of what groupings seemed to be apparent. These groupings became the "categories" to which all descriptors were assigned. This system was extremely useful in determining what types of experts would be needed to review the thesaurus, as well as enabling us to send to the expert reviewer, only those categories that fell within their disciplines. Based, however, on the revisions to the thesaurus that have occurred as a result of the expert review and the prototype cataloging effort (plus the observation of users during the prototype test), these categories have served their purpose and no longer warrant the expense of maintaining them. It is therefore recommended that the categories be discontinued. Users and catalogers will continue to have hard copy thesaurus aids in the form of the alphabetical, hierarchical, and KWIC (keyword in context) or KWOC (keyword out of context) listings.

It is also recommended that catalogers have a better presentation of terms on the screen, i.e., terms in alphabetical order, and have the capability of displaying on the screen the hierarchy to which the descriptor belongs, related terms, definitions, and scope notes.

2.2.10 Abstract

Abstracts were entered for all units that were more than one page in length. Since the ASCII text was not available to catalogers, they had to type in the abstract as it appeared in the unit. If the unit did not have an abstract, and the cataloger wished to use sentences from the introduction, summary, or document text, this information had to be typed in also. This became a very time-consuming process as some abstracts were more than a page long, and the typed input many times contained typographical errors. (Unfortunately, the cataloging software did not contain a spell checker.) To alleviate the situation somewhat, all abstracts were limited to forty screen lines in length. A better long-term solution would be to give the catalogers the capability of "zoning" portions of the ASCII text to pop into the abstract field.

Many abstracts (as well as a few titles) contained non-reproducible symbols, superscripts and subscripts, or short tables. Decisions had to be made on how to handle these for the prototype. For the abstract field, superscripts and subscripts were brought up to the line in the case of chemical notation, tables and charts were eliminated, and Greek letters were spelled out. Complex chemical and mathematical formulas were replaced by the same notation (i.e., ^{^^^}) that was used during ASCII conversion, and most superscripts and subscripts were preceded by the characters <sup> or <sub>. This did not appear to adversely affect the user. Indeed, most users indicated that they would look at the image

for such text, the Technical and Engineering usage group being the one most likely to use documents containing such text.

2.2.11 Document Characterization

The prototype data base contains 2617 records comprising the Site Characterization Plan for Yucca Mountain - Consultation Draft (SCP-CD), references contained therein, files related to its anticipated administrative record, and comment sheets that have been received. The data base contains approximately 100,295 pages, resulting in an average document size of slightly more than 38 pages. The makeup of the data base by type of document is as follows:

	# of Documents	% of Data Base
Correspondence	216	8%
Computer Software	0	0%
Data	49	2%
Governing Documents	171	7%
Legal Materials	3	0%
Manuals	1	0%
Packages	31	1%
Procurement Documents	0	0%
Publications	866	33%
Reports	1282	49%

If the SCP-CD is excluded from the document and page counts, some generalizations can be made about the documents and their corresponding number of pages:

	% of Documents	% of Pages	Avg Page Count
SCP-CD References	72%	81%	44
Admin. Record/Comments	28%	19%	27

If correspondence and maps are deleted from the SCP-CD references, the page count per document is increased by 1. On the other hand, if the correspondence contained in the SCP-CD references is examined as a separate set (43 documents comprising 337 pages), correspondence averages 7.8 pages per document. Given the fact also that some of the references are excerpts of documents, journal articles, conference papers or conference paper abstracts, the average page count for reports would be considerably higher than the 44-page average.

In addition to comment sheets, the administrative record is mainly composed of correspondence, meeting summaries, and meeting materials (handouts,

presentations, discussion papers). This portion also contains some reports and public hearing records which tend to drive up the average page count. On the other hand, many package headers represent collections of documents and do not have any pages themselves. Comment sheets were not entered individually, but in groups, each group covering one subsection of the SCP-CD and one date of discussion (an average of 13 pages per group). (Some comment sheets consist merely of the completed comment form while others had annotated pages from the SCP-CD attached to the form.) If the comment sheets are considered an aberration and are deleted from the page and document counts, the page count per document is only reduced by 1.

Other aspects of document characterization are document size and format, format being especially important in terms of producing searchable text for users. There were very few documents that did not contain some kind of non-standard text (handwriting, tables, graphics, formulas, etc.). Even a simple transmittal letter has a handwritten signature and a letterhead which have to be zoned out for ASCII conversion. In addition, text format could range from standard (10 or 12 pitch) single-spaced pages to triple-column text, could have footnotes at the bottom of the page, numbered lines, fold-out charts (some combining graphics and text), oversize maps with narrative text on a portion of the map, etc. Presenting these kinds of text in searchable ASCII for users can be a costly challenge. It cannot be said that there is a "typical" LSS document. Size of the document can range from a paragraph on a page (conference abstracts) to more than 3000 pages (chapter 8 of the SCP-CD). Text size can range from condensed print to letters several inches in height that can be italicized or underlined. Page size can range from 5 1/2 x 8 1/2 on up to 37 X 56 inches.

Procedures are being written as to how to process these myriad kinds of documents, but it is essential that users comprehend the relationship of the ASCII text to the image, as well as what portions of the page have been converted to ASCII and are therefore searchable. It is recommended that rules be developed that are not be complex and that can be easily communicated to the user.

2.3 Image and ASCII Preparation

2.3.1 Background

The Prototype Development Plan called for the development of a database of approximately 100,000 pages of representative LSS data. Previous experience indicated that this should be of sufficient size from which to obtain realistic user interaction and performance data. It was felt that the development of this database would help establish LSS database requirements as well as provide insight into conversion costs. In particular, the development efforts would provide an opportunity to derive very valuable insight into ASCII text generation, which represents a key (and highly uncertain) cost element of the LSS program. Specifically, the plan called for the gathering of key information regarding the costs and problems encountered in acquisition, cataloging, and

ASCII text generation of representative LSS documents. An important question to be addressed was the extent of editing that would be required to produce an acceptable text database from raw (unedited) OCR data. Prior experience had shown that it could require a significant amount of effort to correct errors introduced during the OCR process. This cleanup activity was felt to be a dominant cost element in the overall ASCII conversion process. The prototype effort was also directed toward understanding the conversion of microform and the use of existing word-processing files.

The Prototype Test plan reiterated the theme of the Development Plan, specifically calling for answers to the following questions:

- o What are LSS documents like?
- o How can data capture be made cost-effective?

2.3.2 Prototype Process and Data Collection

In order to expedite the development of the database, half of the data was processed under subcontract and half internally. The first 50,000 pages were split between two vendors (each processing 25,000 pages). This was done to eliminate a single point failure mode and to provide comparison data on such key issues as throughput and error rate. The remaining 50,000 pages of the database were to be produced by the Nevada Bridge Program Office. The existing Bridge Program equipment in SAIC's Las Vegas Office was to be augmented with the necessary equipment to provide robust production capabilities. This conversion process was to be instrumented to yield detailed data on conversion times and error rates.

Valuable data was garnered from the data produced by vendors. Each vendor used a different process for converting the data, allowing us valuable insight into the effects of certain processes on production efficiency and error rates. One vendor utilized a tightly managed and integrated production facility. This facility handled all aspects of the job including rekeying of those pages which could not be processed efficiently using OCR equipment. The other subcontractor's process was distributed production - imaging and OCR processing taking place at one site in the United States, and all editing and rekeying taking place off-shore. Lessons learned from these two different approaches are included in the paragraphs below.

Unfortunately, problems in obtaining the requested hardware for augmentation of the existing Bridge Program prevented the development of a robust, independent capture system for processing the second 50,000 pages. The alternative prototype hardware/software configuration used for production/instrumented scanning in Las Vegas had significant limitations and therefore did not yield all the data originally intended. NOTE: Valuable lessons were garnered from the use of this configuration, specifically the importance of adequate storage to keep images available and on-line throughout

the conversion process. The inability to do this with the Las Vegas configuration produced many production bottlenecks and inefficiencies. Data regarding hardware and storage requirements was incorporated into the LSS Capture Design and resulting Capture System Specification Document. This situation was exacerbated when, early in the production effort, the Las Vegas Office was forced to relocate to another building. This event precluded the use of the DOE VAX, which acted as the Image and ASCII server, and thereby made it impossible to complete the conversion task in Las Vegas. These exigencies forced a implementation of a contingency plan (the startup of a data conversion activity in SAIC's McLean offices). This contingency plan allowed us to develop the database for user tests but did not provide as much insight into a robust production operation as desired. Both the Las Vegas and McLean operations did yield valuable procedural information which is discussed below.

Initial planning is currently taking place regarding the processing of an additional 30,000 pages of representative LSS data. Procedures and processes will be implemented that, to the extent possible, mimic those in the Capture System Design Document. This effort will be instrumented and closely monitored to yield additional detailed data regarding costs and efficiencies. In addition, a series of controlled tests are planned to compensate for artificialities of this scanning operation in comparison with a genuine capture station. These tests will also allow us to gain data that inherently cannot be obtained from monitoring of production scanning operations. Areas requiring further testing and analysis became evident during the prototype conversion activities. These areas are discussed further in Section 2.3.3.

In addition to the information obtained from the data conversion efforts, data was also derived from the LSS Prototype User Test questionnaires, user observation and debriefing summaries, text retrieval surveys, and SAIC's team observations. This data is also discussed in the following Sections.

2.3.3 Analysis and Observations

2.3.3.1 Quality of Source Documents

The prototype data conversion effort reinforced the strong correlation between source document quality and cost of conversion. We experienced a wide variance in the cost per page - from \$4.00 to as much as \$20.00 per page). This was mainly attributable to differences in the complexity and quality of documents. Cost per page was also affected by start-up costs and changes in operations such as moving from one-shift to two-shift operations. Start-up costs and management costs need to be amortized over large numbers of pages for optimum cost efficiency. Specifically, problems with the data processed in the prototype included:

- o **Poor quality copy**
 - **Blurred and unreadable text due to page curvature problems when copying from books or bound documents**
 - **Words running off a page**
 - **Black borders surrounding smaller than 8 1/2" x 11" pages when the copier reflecting cover was not or could not be closed**
 - **Light copy**
 - **Spray paint appearance of copy due to multiple generation copies or use of a poorly maintained copy machine**
- o **Skewed documents**
 - **Skewed during copy**
 - **Skewed originals**
- o **More than one image on a piece of paper**
 - **Many abstracts on one piece of paper**
 - **Two pages copied on one piece of paper**

The prototype conversion experience supports the need for a better understanding of acquisition costs versus OCR costs. A guideline should be developed for submitters to the LSS regarding desired quality for LSS Documents. This document should set the standards for acquisition based on the trade-off of conversion costs. It should also define what quality of documents will be treated as image-only due to cost considerations. Cost-efficient operation of the capture stations will require that initial processing activities employ specific guidelines for defining rekey versus OCR processing. Based on today's OCR technology, cost-effective operations may call for as much as 40% of the LSS data to be rekeyed. The most efficient subcontractor processed 16,000 of 25,000 pages via OCR; 9,000 were rekeyed. Making the wrong decision can result in unnecessary processing time and increased cost. (See the discussion in Section 2.3.3.6.) Initial processing will also require rules for deciding what documents should be treated as image-only due to the quality of the source document.

2.3.3.2 Accuracy

Based on the recognition that error rate is a significant consideration for the success of full-text retrieval systems, the required accuracy rate for the prototype database was set at 99.8%. This corresponds to two errors per 1000

characters or as many as five misspelled words on a typical LSS page. It is believed that a lower accuracy rate would erode user confidence in the data through missed retrievals. In addition, team members believed that higher error rates would tend to discourage users during searches and thereby make interpretation of user test results more difficult.

High accuracy requirements have significant cost implications. As accuracy approaches 100%, successive improvements become increasingly difficult and expensive due to the need for additional labor hours for editing and proofing cycles. The prototype conversion efforts support that, on average, editing time accounts for as much as 75% of the total processing time.

The Observation and Debriefing Summaries indicate that, in general, users found the accuracy of the prototype database acceptable. Users expressed some tolerance for errors but were quick to point out any misspelled words. At least one user encountered problems with search and retrieval due to misspellings.

While it is clear that a reduction in the required accuracy level would reduce the cost of building the database, it is not clear that a lower accuracy rate would be tolerated by the users. If tools could be provided to users to help overcome problems with search and retrieval due to spelling errors a lower accuracy level may be acceptable. User responses to the Text Retrieval Survey indicate a strong consensus of opinion on the need for text retrieval tools, in particular those dealing with errors such as misspellings.

2.3.3.3 What are LSS Documents Like?

Documents acquired for the prototype conversion effort are believed to be generally representative both in document type and quality of those to be submitted to the eventual LSS database. These documents can be characterized as follows:

- Paper quality on average is acceptable
- Many are duplex documents (two-sided pages)
- Poor photocopies result in skew and unreadable text
- Many contain poor quality print, blurred characters etc.
- Multiple type fonts, ligatures, proportional spacing, and kerning are common
- Many characters are not readily converted to ASCII (i.e., technical notation, formulas and equations)

- Many pages are complex in nature - multiple columns, footnotes, etc. (See Figure 2-1).
- There are multiple page sizes; few are larger than 8 1/2" x 11"
- There were a significant quantity of oversize maps, which were copied in 8 1/2" x 11" panels; (early in the prototype, effort, this process was deemed cumbersome and difficult to use. Since then, a policy was established that the LSS will not accept documents larger than 17" x 22")
- The average page is comprised of approximately 1800 characters
- The average 300 dpi image compressed size using CCITT Group 4 compression is 75 KB

These characteristics of LSS Data were taken into account when defining the LSS Capture Station design. Additional data gathered during the processing of an additional 30,000 pages will be used in finalizing capture processes and in the design of the search and image systems.

2.3.3.4 Formatting Rules and Substitution Codes

The prototype documents presented a number of complex processing issues. These documents contained a significant amount of data which was not readily converted to ASCII due to not being represented in the ASCII character set. In order to capture technical notation, equations, formulas, Greek letters, special symbols, etc., a set of substitution codes were created. There was a question as to whether substitution codes should be created and applied to every unrepresented character. This would allow LSS users that did not have on-line access to images to read all data on a page by using a cross-reference. At the other extreme, some felt that a note of omission in the text was sufficient. Another issue was whether users of the LSS would utilize the substitution codes for search and retrieval.

Aside from the basic format requirements (i.e. preserve the concept of a page and of paragraphs within the page) the following additional format considerations had to be addressed:

- o Is it necessary for the ASCII file to mimic the original source as closely as possible? For example, do paragraph indentions, centered lines, tabs, line lengths, etc. need to be retained line-length. Preservation of format for complex pages with multiple columns presents an entirely different set of problems. For example, see Figure 2-1 which shows a complex page and Figure 2-2 which shows an ASCII representation of that page. Note: In order to preserve the ability to search for the phrase Pleistocene climate, which

Modeling the Climatic Response to Orbital Variations

John Imbrie and John Z. Imbrie

The astronomical theory of the Pleistocene ice ages holds that variations in the earth's orbit influence climate by changing the seasonal and latitudinal distribution of incoming solar radiation (U - J). Because these changes can be calculated with great precision for the past several million years, it is possible in principle to test the theory by comparing the record of Pleistocene climate with a predicted pattern of climatic change. As summarized in the first part of this ar-

climate that are continuous, well correlated, and reasonably well fixed in time. As a result, it has been possible to bypass some of the theoretical problems resulting from our imperfect knowledge of the mechanisms of climatic response by testing the astronomical theory in the frequency domain. For example, it is now clear that a significant part of the observed climatic variance over the past 750,000 years is concentrated in narrow frequency bands near cycles of 19,000,

Summary. According to the astronomical theory of climate, variations in the earth's orbit are the fundamental cause of the succession of Pleistocene ice ages. This article summarizes how the theory has evolved since the pioneer studies of James Croll and Milutin Milankovitch, reviews recent evidence that supports the theory, and argues that a major opportunity is at hand to investigate the physical mechanisms by which the climate system responds to orbital forcing. After a survey of the kinds of models that have been applied to this problem, a strategy is suggested for building simple, physically motivated models, and a time-dependent model is developed that simulates the history of planetary glaciation for the past 500,000 years. Ignoring anthropogenic and other possible sources of variation acting at frequencies higher than one cycle per 19,000 years, this model predicts that the long-term cooling trend which began some 6000 years ago will continue for the next 23,000 years.

tle, early attempts to make such a test encountered both observational and theoretical difficulties. On the one hand, geologists found that their climatic records were discontinuous and difficult to date with sufficient accuracy. On the other hand, climatologists were uncertain how the climate system would respond to changes in the income side of the radiation budget.

During the past 25 years, and particularly over the past decade, the situation has improved dramatically. On the observational side, new techniques for acquiring, interpreting, and dating the pebbliclimatic record—particularly as applied to deep-sea piston cores—have yielded many records of late Pleistocene

23,000, and 41,000 years—as predicted by the simplest and most general (linear) model of climatic response (4-10). If the response is indeed only linear, then statistical analysis suggests that as much as 25 percent of the observed variance is explained by orbital forcing (11, 12). If the response is significantly nonlinear, then the percentage of explained variance may well be larger (13). In any event, there is substantial empirical evidence both in the frequency domain and in the time domain that orbital influences are actually felt by the climate system.

At the same time, significant advances have been made in climate theory. A new generation of radiation-balance models has recently been applied to the ice-age problem with results that support the astronomical theory (14-17). These theoretical developments, combined with advances in paleoclimatology, indicate that there is an opportunity to

make a fundamental shift in research strategy (17). Specifically, we should aim to identify mechanisms by which different parts of the climate system respond to changes in radiative boundary conditions. The importance of this opportunity is that both the temporal and the spatial structure of these changes can be specified exactly (18, 19). Except for studies of the annual cycle, we know of no other problem in climate dynamics where the primary external forcing terms are known so precisely.

This opportunity can be exploited most effectively by comparing geological observations with predictions derived from physically based models of climate. In the second part of this article we review the available models and distinguish between two types: an equilibrium model of the form $y = f(x)$, where the climatic state (y) has come to equilibrium with the fixed orbital boundary condition (x), and a differential model of the form $dy/dt = f(y, x)$, where integration yields a history of climatic response to changing boundary conditions.

In the third section of this article, we recommend a strategy for developing differential models that are physically motivated yet simple enough so that their full explanatory powers can be determined by comparing a wide range of model variants with geological observations. Applying this strategy in the fourth section, we develop a differential model of the form $dy/dt = (1/T_1)(x - y)$. Here, the input x is a single function of time representing the orbital forcing, y is a measure of the total volume of land ice, and T_1 is a pair of constants at our disposal. The output of this model compares favorably with the geological record of the past 250,000 years. Thus, the model's prediction for the next 100,000 years is useful as a basis for forecasting how climate would evolve at orbital frequencies in the absence of anthropogenic disturbance. For intervals older than about 250,000 years, however, the match between actual and model climates deteriorates with age. In a final section, we compare existing differential models, suggest some improvements that could be incorporated in future models, and discuss the fundamental questions of climatic sensitivity and predictability.

Astronomical Theory of Climate

Orbital control of irradiation. If the solar output is assumed to be constant, the amount of solar radiation striking the top of the atmosphere at any given latitude and season is fixed by three ele-

John Imbrie is Henry J. Dobson professor of geology, Brown University, Providence, Rhode Island 02912. John Z. Imbrie is a National Science Foundation postdoctoral fellow in the Department of Physics, Harvard University, Cambridge, Massachusetts 02138.

Figure 2-1
Sample of Original

Modeling the Climatic Response to Orbital Variations

John Imbrie and John E. Imbrie

Summary. According to the astronomical theory of climate, variations in the earth's orbit are the fundamental cause of the succession of Pleistocene ice ages. This article summarizes how the theory has evolved since the pioneer studies of James Croll and Milutin Milankovitch, reviews recent evidence that supports the theory, and argues that a major opportunity is at hand to investigate the physical mechanisms by which the climate system responds to orbital forcing. After a survey of the kinds of models that have been applied to this problem, a strategy is suggested for building simple, physically motivated models, and a time-dependent model is developed that simulates the history of planetary glaciation for the past 500,000 years. Ignoring anthropogenic and other possible sources of variation acting at frequencies higher than one cycle per 19,000 years, this model predicts that the long-term cooling trend which began some 6000 years ago will continue for the next 23,000 years.

The astronomical theory of the Pleistocene ice ages holds that variations in the earth's orbit influence climate by changing the seasonal and latitudinal distribution of incoming solar radiation (1-3). Because these changes can be calculated with great precision for the past several million years, it is possible in principle to test the theory by comparing the record of Pleistocene climate with a predicted pattern of climatic change. As summarized in the first part of this article, early attempts to make such a test encountered both observational and theoretical difficulties. On the one hand, geologists found that their climatic records were discontinuous and difficult to date with sufficient accuracy. On the other hand, climatologists were uncertain how the climate system would respond to changes in the income side of the radiation budget.

During the past 25 years, and particularly over the past decade, the situation has improved dramatically. On the observational side, new techniques for acquiring, interpreting, and dating the paleoclimatic record-particularly as applied to deep-sea piston cores-have yielded many records of late Pleistocene climate that are continuous, well correlated, and reasonably well fixed in time. As a result, it has been possible to by-pass some of the theoretical problems resulting from our imperfect knowledge of the mechanisms of climatic response by testing the astronomical theory in the frequency domain. For example, it is now clear that a significant part of the observed climatic variance over the past 730,000 years is concentrated in narrow frequency bands near cycles of 19,000, 23,000, and 41,000 years-as predicted by the simplest and most general (linear) model of climatic response (4-10). If the response is indeed only linear, then statistical analysis suggests that as much as 25 percent of the observed variance is explained by orbital forcing (11, 12). If the response is significantly nonlinear, then the percentage of explained variance may well be larger (13). In any event, there is substantial empirical evidence both in the frequency domain and in the time domain that orbital influences are actually felt by the climate system.

At the same time, significant advances have been made in climate theory. A new generation of radiation-balance models has recently been applied to the ice-age problem with results that support the astronomical theory (14-17). These theoretical developments, combined with advances in paleoclimatology, indicate that there is an opportunity to make a fundamental shift in research strategy (17). Specifically, we should aim to identify mechanisms by which different parts of the climate system respond to changes in radiative boundary conditions. The importance of this opportunity is that both the temporal and the spatial structure of these changes can be specified exactly (18,19). Except for studies of the annual cycle, we know of

Figure 2-2

Sample of ASCII Conversion

no other problem in climate dynamics where the primary external forcing terms are known so precisely.

This opportunity can be exploited most effectively by comparing geological observations with predictions derived from physically based models of climate. In the second part of this article we review the available models and distinguish between two types: an equilibrium model of the form $y = f(x)$, where the climatic state (y) has come to equilibrium with the fixed orbital boundary condition (x), and a differential model of the form $dy/dt = f(y,x)$, where integration yields a history of climatic response to changing boundary conditions.

In the third section of this article, we recommend a strategy for developing differential models that are physically motivated yet simple enough so that their full explanatory powers can be determined by comparing a wide range of model variants with geological observations. Applying this strategy in the fourth section, we develop a differential model of the form $dy/dt = (1/T\langle ii \rangle)(x - y)$. Here, the input x is a single function of time representing the orbital forcing, y is a measure of the total volume of land ice, and T is a pair of constants at our disposal. The output of this model compares favorably with the geological record of the past 250,000 years. Thus, the model's prediction for the next 100,000 years is useful as a basis for forecasting how climate would evolve at orbital frequencies in the absence of anthropogenic disturbance. For intervals older than about 250,000 years, however, the match between actual and model climates deteriorates with age. In a final section, we compare existing differential models, suggest some improvements that could be incorporated in future models, and discuss the fundamental questions of climatic sensitivity and predictability.

Astronomical Theory of Climate

Orbital control of irradiation. If the solar output is assumed to be constant, the amount of solar radiation striking the top of the atmosphere at any given latitude and season is fixed by three ele-

John Imbrie is Henry L. Doherty professor of oceanography, Brown University, Providence, Rhode Island 02912. John Z. Imbrie is a National Science Foundation predoctoral fellow in the Department of Physics, Harvard University, Cambridge, Massachusetts 02138.

SCIENCE, VOL. 207, 29 FEBRUARY 1980

0036-8075/80/0229-0943\$02.00/0 Copyright ^^^ 1980 AAAS

943

Figure 2-2

Continued

2-22

splits between column one and column two, proper ordering of the columns and repositioning of the insert and footnote was required.

- o Should tables be converted to ASCII or treated as image data? What is the definition for a table? Should tables with "readable" text be converted to ASCII? (See Figure 2-3). If so, is it necessary to retain the format of the table or only capture the textual content for full-text searching?

A set of formatting rules and substitution codes were generated for the processing of prototype data. This set of rules called for preservation of "relative" format whenever possible, as well as the rigorous application of substitution codes for converting technical data within text.

In order to gain insight into format requirements, the prototype database included two types of formatted text:

- o Text that preserved the concept of a page and a paragraph but did not mimic original format [text was brought to the left margin]
- o Text that replicated the original format whenever possible, (for example, paragraph indentions, centered lines, etc. closely matched the original source.)

Immediately, after the start of processing it was clear that the complex nature of LSS data resulted in many variations in interpreting these rules. Consistency was very difficult to achieve. These various interpretations resulted in significant editing and quality control time. It was immediately apparent that formatting and substitution code requirements would have a significant impact on the cost for converting LSS data. These results indicate that it is imperative to have clear and concise rules which leave no room for interpretation. Another simple, but overriding rule is to minimize exception processing.

Conversion efforts support that format cannot be preserved for many complex documents and that reordering by use of a zoning process is often required. These reordering decisions can greatly inflate the cost of conversion when scanning operators are required to make these decisions as well as to zone the documents for OCR. Processing times for scanning and quality control can be better managed if this task is completed during the initial processing stage by more senior personnel who understand search strategies and requirements. This decision also facilitates a more efficient process for indicating the order within documents if they are to go off-site for rekeying.

The use of macro keys for the keying of substitution codes resulted in a more efficient process.

Table 1. Listing of all independent variables defined in the study of climate and available for stepwise regression predictions of precipitation and temperature. Each variable in the second group is computed for both February and August. Thus, for example, there is ELMUPF and ELMUPA. In group three all months may be available. Thus there is a TJAN, TFEB --- and CUPJAN, CUPFEB, etc.

	<u>Name</u>	<u>Variable</u>
1.	LAT	Latitude of Point
	LONG	Longitude of Point
	ELEV	Elevation of Point
	SLOPES	Slope to South from Point to Next Adjacent Point of Grid
	SLOPEN	Slope to West from Point to Next Adjacent Point of Grid
2.	ELMUP	Elevation of Maximum Upwind Point
	DISMUP	Distance to Maximum Upwind Point
	ELLOP	Elevation of Lowest Point Between
	DISLOP	Distance to Lowest Point Between
	ELEDROP	Elevation Drop (Maximum Upwind - Point Elevation)
	ELEGAIN	Elevation Gain (Lowest Point Between - Point Elevation)
	LNDIMUP	LN (Distance to Maximum Upwind Point + e)
	LNDILOP	LN (Distance to Lowest Point Between + e)
	DECADRF	Elevation Drop/LN (Dist. to Max. Upwind Point + e)
	DECAGAN	Elevation Gain/LN (Dist. to Lowest Point Between + e)
	DECAMAX	Elevation of Maximum Upwind Point/LN (Dist. to Max. Upwind Point + e)
	DECAMIN	Elevation of Lowest Point Between/LN (Dist. to Lowest Point Between + e)
	SST	Sea-Surface Temperature
	DISCOS	Distance to the Coast (Along Wind Vector)
	LNDICOS	LN (Distance to the Coast + e)
	DECSST	Sea-Surface Temperature/LN (Distance to the Coast + e)
	3.	T
CUP		Cube Root of Monthly Mean Precipitation (if previously predicted)

Figure 2-3
Sample of Original Table

Analysis of the responses to the User Questionnaire (See Appendix C) and the Observation and Debriefing summaries indicate the following:

- o Although upper and lower case letters are not required for searching, preservation of capitalization is preferred when reading the ASCII files.
- o Variation in page format between the ASCII file and the original document does not erode user confidence in the database.
- o Word-wrap should be avoided.
- o Substitution codes would not be searched.
- o ASCII conversion of equations and formulas by use of substitution codes would not be necessary if the image is available. Indication that something was omitted (for example the use of the omnibus substitution code <technical notation> would be of value.)
- o Text within tables is of value for searching. (Users would like to see format retained for textual tables, and tables of contents.)
- o Capture of the literal page number is required.
- o Initial guidelines regarding, boldface, italics, footnotes, superscripts, subscripts, etc. appear to be reasonable and accepted by users.

A set of controlled tests are being developed to provide quantitative data regarding the cost for implementing various levels of formatting and substitution codes. This data along with user requirements will provide a basis for establishing the most cost-effective rules for eventual LSS data conversion activities.

2.3.3.5 Image Scanning

Based on knowledge of the industry and prior experiences with users of image systems, images for the prototype were scanned at a resolution of 300 dpi x 300 dpi. The LSS Prototype User Tests support that this is an acceptable scanning resolution. Grayscale was not considered to be a requirement. This also proved acceptable to users of the prototype system. (See Section 4.2.3.2 for additional comments and analysis.)

One-to-one correlation of the LSS images and corresponding ASCII pages will be a requirement for the LSS database. Each of the subcontracted vendors took a different approach to achieving this goal. Problems were encountered by one of the vendors who used a disjoint process for the scanning of images and the

processing of matching ASCII files. This provided important insight into the cost ramifications of maintaining one-to-one correlation of these files. This vendor had a particular problem with the scanning of duplex (two-sided) pages. Many pages were scanned out of order and/or upside-down which resulted in an incorrect order for the ASCII file. While these types of process errors should be expected (this process has many of the same problems that are encountered when making a copy of a two-sided document) quality review of the other vendor's product did not show this to be a problem.

The vendor having problems in maintaining one-to-one correlation did not effectively identify documents at the page level. This vendor did not mark all pages of a document with its logical page number prior to processing. Therefore, when human errors occurred during scanning (which can be expected) causing problems with the order of pages, it was a very labor-intensive and sometimes impossible task to reorder the document correctly. This was exacerbated by the fact that images and ASCII files were created at different locations resulting in an ineffective quality control process. Error correction was logistically cumbersome, resulting in an inability to achieve the required accuracy level.

The other vendor had meticulously marked all pages with a unique identifier indicating logical page number prior to any processing. This unique identifier was included in the image header and was translated into ASCII as part of the ASCII file. Review of production efforts showed that this vendor also encountered problems with page order, especially when processing duplex documents. However, the system in place that uniquely identified every page of each document allowed for order correction with minimal amounts of effort. When a document was scanned out of order it could easily be reordered electronically by review and reorder of a directory. It was also observed that since all aspects of the conversion process took place at one site, efficient error correction was easily achieved. SAIC's quality review of this subcontractors work showed a consistent achievement of the 99.8% accuracy specification.

As a system result of the above analysis, a page orientation was adopted for the LSS Capture design. Each page of an LSS document will be bar-coded with the unique LSS accession number and logical page extension. This identification will take place during initial processing. This fundamental step will significantly reduce the amount of labor and software that could otherwise be required for keeping track of page order. This is anticipated to result in significant cost savings for conversion of LSS documents. This procedure will be further tested and refined during the processing of another 30,000 pages of data.

This analysis also led to a better understanding of initial processing requirements which are discussed below and in Section 2.1.

2.3.3.6 Preprocessing

Prototype conversion efforts resulted in the establishment of a comprehensive preprocessing requirement. It was quickly established that LSS data called for a number of complex processes and analytical decisions which needed to be made on a page-by-page basis. Production observations supported that these important decisions and processes were best accomplished by the use of a highly-trained preprocessing staff, familiar with LSS data, versus placing these burdens on production personnel. Correct preprocessing of documents can result in both improved overall efficiencies and a higher accuracy rate. The experience gained indicates that preprocessing responsibilities should include:

- o OCR or Rekey Decision - A cost-effective conversion effort will require both the use of OCR technologies and the rekeying of text. It is clear that using OCR for all data or rekeying of all data can inflate the cost of conversion. These rekeyed decisions should be made at the page level. Preprocessors should have a set of guidelines to assist them in making these decisions. These guidelines should be amended based on feedback from the OCR editing process.
- o Zoning Decisions - Prototype conversion efforts support that processing times for scanning and quality control can be better managed if zoning instructions for a page are created during the initial processing stage by highly trained personnel.
- o Processing Instructions - Determination of unreadable text, what parts of a page to image, etc. should be accomplished at initial processing and instructions passed to production operators.
- o Logical Page Numbering - See Section 2.3.3.5. This is believed to be a fundamental requirement for efficient and cost-effective processing.

2.3.3.7 Editing

The most efficient subcontractor-provided data shows editing time to be 74% of the total processing time; data from the in-house conversion operation indicates editing time to be 77% of the total processing time.

There are two types of errors generated during the OCR process:

- (1) Unreadable characters - If the OCR has trouble recognizing a character, it inserts a marker. These markers must be reviewed, compared to the original text, and corrected by an editor.

- (2) **Misread characters** - When the OCR machine misreads a character and inserts another character in error, a misspelled word most likely occurs. OCR devices also add random spaces and letters when encountering stray marks on a page. These random errors and spaces can cause misspelled words as well as retrieval errors when conducting a proximity search. Misread characters are corrected by the running of a spell check and/or a visual proof of the document by an editor.

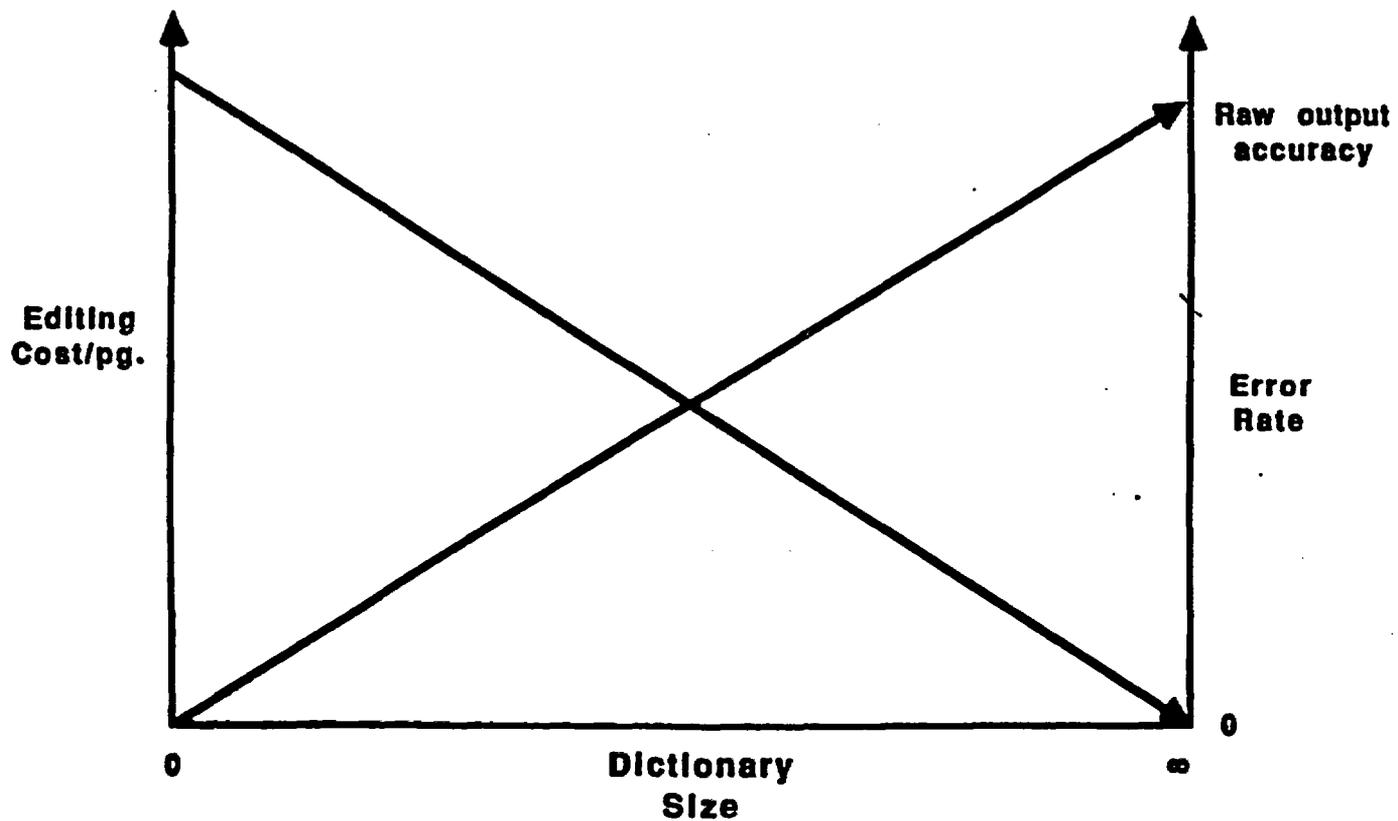
Editing in the case of LSS data conversion also includes insertion of substitution codes, application of formatting rules, spell checking, visual proofs, and quality control processes. The prototype effort supports that as part of the editing process, visual proofs of some documents and random checks for one-to-one correlation of images and ASCII files are required to reach 99.8% accuracy for the database. Quality control should also include random spell checks. Rekeyed documents also require these quality control processes, to assure a 99.8% accuracy rate.

Correct implementation and control of the size and contents of the dictionary used for spell check can help minimize costs. Figure 2-4 shows that editing costs are at their highest when there are few words included in the dictionary. When all possible words are included in the dictionary, the spell check is ineffective and accuracy suffers. It is important to understand what is the optimum dictionary size that will minimize editing time and maximize the accuracy of output. Prototype efforts indicate that the LSS spell-check dictionary will be large (The prototype dictionary contained on the order of 210,000 words - approximately 100,000 of these terms are LSS specific). We do not yet understand the point of optimization for this editing tool. During the processing of the 30,000 pages we will continue to address this issue by observation, statistical monitoring, and controlled tests. We also need to gain an understanding into the use of multiple (subject-oriented) dictionaries to achieve additional cost efficiencies. This will also be addressed during the running of controlled tests.

A decision was made that, for purposes of the prototype conversion effort, spelling mistakes in original documents would be corrected. The cost of editing goes up dramatically if errors in the original text must be maintained. This increased cost is due to the labor-intensive job of verifying all misspellings against the original copy prior to correction. Correction of errors found in original documents also improves retrieval capability. Users of the prototype database expressed no adverse reaction to this decision.

Cost of editing is affected by changes in line-length. Proofing takes more time if the lines have been wrapped. For cost-efficiency, line-length should remain constant with the original whenever possible.

At the end of the editing phase, tabs and macros need to be replaced with the appropriate character strings because their definitions are transitory.



Dictionary Effects

Figure 2-4
Dictionary Size Effects
2-29

2.3.3.8 Microform and Word-Processing

Studies and tests conducted during the prototype effort support the feasibility of converting microform documents. However, due to the page orientation for the capture process, the overriding issue became how to effectively label pages and documents for OCR processing. Due to the small percentage of LSS documents that need to be converted from microform, it was believed that the most cost-effective solution would be to minimize exception processing and print all microform as hardcopy and follow standard operations for capture. In the course of the past year, there have been significant improvements in equipment for direct digitization of image from microform. These rapid changes in technology may allow for an alternative cost-effective approach for capture of LSS microform.

Our studies and analyses of utilizing existing word processing files indicate that this practice can result in equal or greater costs than creation of the ASCII file from the submitter's image. This is due to the amount of labor required to assure the integrity of the database, i.e., does the word-processing file match the submitters image? To minimize costs, the Capture Station design requires that the submitter assure the integrity of submitted data. Additional data regarding how to effectively link images and word processing files, as well as perform effective quality control, will be gathered during the processing of the last 30,000 pages.

2.3.4 Summary of Lessons Learned

In summary, the key lessons learned are the following:

- o The accuracy of the prototype database (99.8%) is acceptable.
- o Variations in page format from that of the original document are acceptable.
- o Procedures for acquisition will have to be developed that balance acquisition costs against text conversion costs.
- o It is believed that page level control for capture operations will reduce overall costs.
- o A complex, comprehensive initial processing effort will be required to minimize costs.
- o Specific guidelines need to be developed to support cost-effective OCR versus rekey decisions.

- o A cost-effective operation may result in as much as 40% of the data being rekeyed
- o Spell-check dictionaries will be large (in excess of 250,000 words).
- o Editing costs may be reduced by use of an optimized spell-check dictionary and minimizing requirements for the use of substitution codes and formatting rules.

A number of key decisions for LSS data capture operations are yet to be made. These decisions will need to be reflected in very detailed and specific operating instructions for data capture. These instructions will reflect numerous tradeoffs between conversion costs and retrieval quality. Tradeoffs that need to be addressed include:

- o the cost of highly accurate data conversion versus the cost of implementing more powerful text retrieval tools
- o acquisition costs versus text conversion costs
- o the cost of providing images versus implementing formatting rules and substitution codes.

Specific system requirements based on these lessons are discussed in Sections 5 and 6 of this document.

3.0 DATA PREPARATION AND LOAD

The lessons learned regarding the preparation of the cataloged headers, ASCII text and the images for loading into the Prototype retrieval systems are discussed in this section. First, the processes involved are described, emphasizing the key factors that affect quality and resources required. Second, resources required for the indexing and loading headers and ASCII text are analyzed. The questions from the Prototype Test Plan addressed are:

What resources are required for selected operations?

- o Data base indexing and loads
- o Overhead for storing indexes

What resources need to be allocated to loading large volumes of data?

3.1 Data Preparation Process

The data preparation process involved several steps and was different for the two systems, remote and local. The steps associated with the remote system, which had both images and partitions, more closely match the projected LSS load process.

Header preparation. The cataloged headers were extracted from the Quality Control header data base. The headers were written to an ASCII file with the beginning of each header marked with the accession number, document type, and date. Each field's label was placed in the file. Long fields such as the title and abstract were set to line wrap on word boundaries.

ASCII text collection. All the ASCII text was collected and organized on the DOE provided computer that supported the remote system. The ASCII text was received both on magnetic tape and high density floppy disks. The ASCII text from the floppies was uploaded. After all the ASCII text was collected and organized, a copy of the ASCII text was downloaded to the local system 386 micro computer using an Ethernet LAN.

Assign partition and load order. The document type and date were used to assign the documents to a partition. Then, the documents for each partition were sorted into descending date order, so the most recent documents in a result set would display first. This process only applied to the remote system.

Context unit determination. Each text retrieval DBMS had its own context units for proximity searching. To support those units, the ASCII text had to be marked. For the local system, one of the context units was paragraphs. A text tag was placed at the start of each paragraph. Developing a rule to identify paragraphs was difficult, with title pages, table of contents, list of

figures and imbedded tables causing the most problems. The context unit used in the remote system was the sentence. The load utility for the remote system had its own rules for determining end of sentence, but it was found that some ASCII text cleanup helped. Again, entire title pages and table of contents, often were identified as one sentence.

Image index extraction. This process built an index from the optical disks that contain the images. The index related the optical disk location of each image to its corresponding ASCII text page.

Combine headers and ASCII text. For the remote system, this process involved appending the ASCII text to the header and placing a text label at the beginning of the text. Image pointers were inserted at the end of each page. ASCII text lines were adjusted to be less than 80 characters in length to support the screen widths of the remote system display stations. Some of the editing processes had used tabs to maintain alignment, but the data base systems would not handle tabs, so tabs were replaced by spaces. Except for the emplacement of the image pointers, the process was similar for the local system.

Dividing large documents. The text retrieval DBMS used for the remote system was furnished by DOE and was used in the configuration it had been installed for general use. That installation had a document size limit that was too small for some of the chapters from the SCP and several other large documents. For the remote system, those documents were subdivided. Headers were created for those documents.

Creating a load file. After the header and ASCII for each document had been prepared and merged, all documents for a given partition were appended into a single load file. For the local system, the output of the header and ASCII text combining process was a single load file.

3.2 Data Base Load

The data load process involved the following data volumes:

Headers - 2748 documents, 5.5 million characters
ASCII text - 78,000 pages, 145 million characters

Local System Load. The load utility provided with the text retrieval DBMS was used to load the data base. The input data, consisting of 132 million characters, the data base, and index, had to fit into the hard disk storage of the local system. There were no partitions on the local system, so the headers and ASCII text were loaded into a single data base. The load was done as one continuous process which ran about 13 hours. The resulting data base size was

about 106 million characters and the data base index was 79 million characters.

Remote System Load. Since the remote system computer was not dedicated to the prototype effort and the data base load required most of the computer's capacity, the loads were run late at night or over weekends. Each partition was loaded separately. The load process required a lot of disk space to support the sort process used in the index build portion of the load. To stay within the disk space available, several of the large partitions entailed more than one load. The times required for the loads are shown below and graphically in Figure 3-1.

Partition	Load File Size Millions of characters	Load Time in hours
Correspondence	.7	.5
Publications	37.7	11.0
Governing Documents	11.8	3.0
Packages	.05	.5
Manuals	.4	.5
Reports 1	46.0	13.0
Reports 2	32.9	10.0
Reports 3	20.7	6.0

3.3 Observations and Conclusions

The data preparation process is tedious but nevertheless critical to the success of the load process. The establishment of the context units is the most critical preparation step affecting retrieval. In the remote system, the entire context unit (a sentence) where a hit occurred was highlighted. When users found a hit in the title or Table of Contents, the entire section highlighted, since it often looked like a sentence to the load utility. To help eliminate this problem, during clean up, some marking of the ASCII text to identify title pages, tables of contents, list of figures and tables would be very helpful. Another approach, would be to use a set of "AI-like" rules. The context units to be used in the LSS must established before the specific processes, marks, or rules can be developed. Since there are no standards in the text arena, each text DBMS has its own set of context units. This raises the question whether or not specific context units should be a requirement in the specifications for the LSS text DBMS. If not, then the specific contexts units cannot be established until the LSS Text DBMS is selected.

The use of tabs must be prohibited during the rekey, edit and clean up processes. Only spaces must be in the ASCII text presented for data load. The establishment of the line length also should be done as part of the editing and clean up processes. Given that the level I workstations will likely have only

an 80-character wide display, it is recommended that 80 characters, which was used in the prototype, be established as the LSS line width.

File naming conventions for the headers, ASCII text, and images are required to ensure that the three are properly linked at load time. The conventions, based on the document accession number and page, used during the prototype worked well. However, having the accession number in a standard place in the header and ASCII text files would provide for fool-proof check. For the prototype, the image index included the volume name of the optical disk containing the image. In the LSS the actual volume name of where the image will be stored will not be known at ASCII text load time, since the image loading will be done in parallel. Thus the link between the LSS search and image systems must be via an image name or label that is independent of where the image will be stored.

The resources required for the data base load are significant. The resources needed include disk space required to build the full text index, the central processor unit (CPU) time, and the input/output (I/O) time to and from the disk. As the data base gets larger and larger, the index build time will continue to increase, as will the disk space required for the associated sorting process. Figure 3-1, developed from the remote system load data, gives some indication of the magnitude of the problem. Additional data for other text DBMS packages will be obtained from surveys of users.

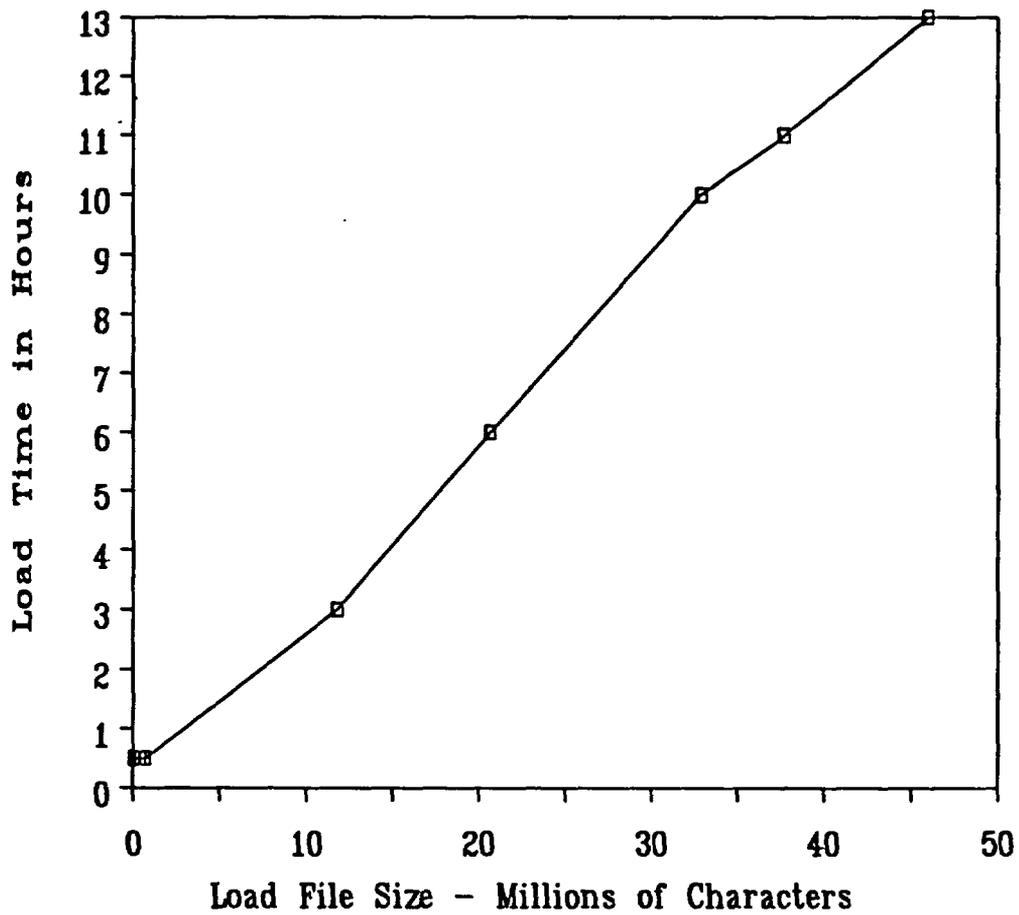


Figure 3-1
Load Times versus Load Size
(Remote System)

4.0 ANALYSIS OF USER TESTS

This section presents the analysis of the data collected during the user test. Data was collected in five ways: 1) computer logs that recorded keystrokes of the user, 2) notes recorded by a user observer, 3) notes that the user has recorded on the question sheets, 4) a user questionnaire, and 5) user responses to debriefing questions. The statistics generated from this data are presented in the four Appendices to this report as follows:

- 1) Appendix A - LSS Prototype User Session Statistics, Remote System, which provides the statistics generated from the remote system computer logs.
- 2) Appendix B - LSS Prototype User Session Statistics, Local System, which provides the statistics generated from the local system computer logs.
- 3) Appendix C - LSS Prototype User Exit Questionnaire Statistics, which provides the statistics generated from the questionnaires.
- 4) Appendix D - LSS Prototype Sample User Logs, which contain listings of sample computer logs generated for both the remote and the local systems. Because of their volume, the total logs are not included with this report, but they will be maintained in the LSS Prototype files for future reference.

Table 4-1 presents a summary of user test statistics for the remote and local systems derived from Appendices A and B.

Section 4.1 describes the user test and Section 4.2 presents the results of the analysis.

4.1 User Test Design

The user test was designed to provide both quantitative and qualitative data that would assist in addressing design issues of both the Capture and the Search and Image Systems of the LSS. Ideally, the test would provide an understanding of different users' perceptions of the information contained in the data base and of the strategies they employ to retrieve that information. Such an understanding will affect the storage of information in the header, the conversion of information to searchable text, the system resources required to support user search and retrieval needs, and the design of the user interface.

The mechanism used to collect the data was a two-day test session for each user. This test session was divided into a half-day of training, two half-day sessions of data base searching, and a half-day of debriefing. Since the

Table 4-1 User Tests Summary Statistics

USER TESTS SUMMARY STATISTICS			
CHARACTERISTIC	LOCAL SYSTEM	REMOTE SYSTEM	TOTAL
Avg time/question (min)	23.2	31.5	28
Number of logged sessions	266	353	619
Number of questions	244	333	577
By user:			
TE	71	133	204
RL	81	81	162
MA	39	49	88
IN	42	55	97
PI	11	15	26
Number of queries	1358	2256	3614
By user:			
TE	352	841	1193
RL	497	691	1188
MA	226	283	509
IN	216	304	520
PI	67	137	204
Number of queries/question	5.57	6.77	6.26
By user:			
TE	4.96	6.32	5.85
RL	6.14	8.53	7.33
MA	5.79	5.78	5.78
IN	5.14	5.53	5.36
PI	6.09	9.13	7.85
Number of displays	807	1030	1837
By user:			
TE	209	413	622
RL	322	322	644
MA	140	134	274
IN	97	112	209
PI	39	49	88
Number of displays/question	3.31	3.09	3.18
By user:			
TE	2.94	3.11	3.05
RL	3.98	3.98	3.98
MA	3.59	2.73	3.11
IN	2.31	2.04	2.15
PI	3.55	3.27	3.38
View size			
Average	13.2	9.1	10.9
Most frequent	1	1	1

responses of a somewhat experienced user would provide the most useful data, a "crash course" was given to each user on the system to which the user had been assigned, either the remote system, the local system, or the image system. In addition, each system was manned by a system expert who functioned as an "online manual" or "help screen". The system expert did not provide search strategy assistance to users, but merely answered questions regarding how to use the system to perform a certain function (e.g., view a result set, display a list of terms, go to the end of a document) or pointed out features of the system that would assist the user in accomplishing his strategy. This concept appears to have been successful since comparison of the statistics indicate, for example, only a moderate (20%) increase in the number of queries issued in Day 2 compared with Day 1. (Day 1 being all of the first half-day sessions and Day 2 being all of the second half-day sessions.)

The test groups were composed of a mix of user types based on the usage groups identified in the Licensing Support System Preliminary Needs Analysis: Technical and Engineering, Regulatory and Licensing, Management and Administrative, Public and Public Information, and Intermediary. The number of users for each group was roughly proportional to the usage of the LSS expected by each group.

Each user was given a package containing six questions prefaced by a scenario. The scenario presented a situation to the user and told him his role in the situation. Each scenario was tailored to the type of user that would be playing the role. For example, a Technical and Engineering user was asked to play the role of a geologist investigating faults in the Yucca Mountain area, while the role of a Management and Administration user was to be as a staff member of a Congressional Affairs Office drafting responses to Congressional questions. The questions for each scenario were based on information that a person in that role would be likely to seek. Each question was designed to elicit one of three types of responses: a search for a specific document, a search for specific information that may be contained in one or more unknown documents, a search for all information on a particular point. Each scenario contained six questions, two of each type. A user completed all questions for a scenario before he was given another one. Users were encouraged to write their strategies, comments, and any other useful data directly on the question sheets. In addition, a silent observer recorded user questions, comments, and behavior. It was stressed to the users that the team was not interested in the answers to the questions themselves but in the strategies and approaches used to find a satisfactory answer. Despite this message, conversation with some users revealed that they thought of the exercise as a test of themselves rather than a revelation of their behavior on a test system.

The debriefing was done in two parts: a written questionnaire that was completed by each user independently and an private interview conducted by an experienced debriefer. All questionnaire and interview questions were coded to

the original list of questions contained in the LSS Prototype Test Plan to assist in data analysis.

All of these data-gathering mechanisms provided the analysis team with cross-checks on three important points: how the user actually retrieved information, how the user perceived that he obtained the sought-for information, and how the user would like to retrieve such information. The following section presents the results of the data analysis under four main topics: header, full text, images, and general issues.

4.2 Analysis Results

This section presents the results of the data analysis. Each subsection discusses the pertinent questions from the LSS Prototype Test Plan, analyzes applicable quantitative and qualitative data, and presents conclusions and recommendations.

4.2.1 Header

Organizing information into fields in a header can be a powerful tool for a user for both searching and presenting information. For example, a user searching for a specific name can limit the search to an author field. Similarly, a user can present the retrieved data in alphabetical order by author using that same field. The full text of a document in searchable form is another powerful tool for both searching and retrieving. For example, a user can search for every occurrence of a particular name or topic. The user can also retrieve the text of a document without going to a library, bookstore, or government agency. A system which combines header fields with the searchable full text of a document should give users the best of both worlds. However, combining full-text capability with a header can change the usefulness of some of the standard header fields, depending on the size of the database. A small data base may not need a descriptor or keyword field since the number of full-text "hits" on a particular topic is not likely to be greater than the number a user is willing to browse through to find the desired information. However, for a large data base (such as the LSS) sophisticated descriptors in conjunction with full text may be essential to reduce a large set of "hits" to a browsable number.

Questions in the LSS Prototype Test Plan address the issue of an optimum header design which will contain those fields useful for retrieval and display and eliminate the expense of capturing and storing data in fields that are rarely used, given the advantage of full text. These questions include:

What header fields do users use and how will they use them?

What depth of descriptor assignment best helps users?

What special tools will be beneficial to searchers?

What search strategies will users employ?

How does the system encourage or discourage effective user strategies?

To some extent, the header fields used for querying and display were a function of the scenario question the user was answering. Training, user biases, user type, and peer pressure also influenced the use of header fields. In order to provide as many access points to the information as possible to evaluate field usage, the header contained a large number of fields. In addition, some of the fields, such as Publication Data and Special Class, were the equivalent of two or more fields. To help determine the depth of descriptor assignment, descriptors were assigned for both major and minor subjects addressed in the document. Quantitative data on field desirability was obtained from both the questionnaire and the session logs; qualitative data was extracted from the questionnaire and the debriefing interview.

4.2.1.1 Header Fields

Fields can serve two purposes: to assist users in retrieving information and to display to users the information that has been retrieved. A field that is useful for one purpose may or may not be useful for the other. For example, a user probably would not want to search on page count, however, page count can be valuable in display since it tells the user, among other things, the number of pages that will be printed if the print command is executed. One of the questions in the user questionnaire asked the users to rank the usefulness of all prototype fields first for searching value and then for displaying results. Users were asked to select both the five most valuable and the five least valuable. The percent of users selecting a field as most or least valuable is shown in Table 4-2.

TABLE 4-2
FIELD USE (QUESTIONNAIRE)

Most Valuable		Least Valuable	
<u>SEARCHING</u>			
Title	75%	Media	64%
Author Organization	64%	Document Condition	59%
Descriptors	61%	QA Level Code	55%
Date	50%	Special Class	45%
Abstract	43%	Page Count	41%
Author	41%		
<u>DISPLAYING</u>			
Title	80%	QA Level Code	50%
Abstract	64%	Media	50%
LSS Accession Number	57%	Document Condition	45%
Author	43%	Special Class	43%
Date	41%		
Author Organization	36%		

Note: only fields with values greater than 35% are listed.

It can be seen from this table that (based on user perception), the title, abstract, author organization, date, and author are the most useful fields with descriptors and accession number running a close second.

The least useful fields are consistent across the board and a little harder to analyze. This may be the result of users not understanding the field or of the field not being useful for this test. For example, all documents were part of the SCP-AR; this essentially provided users with a pre-search cut of the data base, i.e., all records that belong to Special Class SCP-AR. QA Level Code is the code assigned to a document that furnishes evidence of the quality and completeness of data and activities affecting the quality of the repository. Either users did not understand this field or, since this designation appears on the document itself, there may be little need to have the information in a field. Media tells the user the media of the original document. For the prototype, all media was paper. However, in the LSS this field might have such entries as physical sample, computer code, videodisc, etc.; this might be valuable information for those items that have no image or ASCII text. Document condition is a similar field, telling the user if the document is illegible, has missing pages or attachments, is unsuitable for some reason (unscannable media, oversize document, etc.), or has marginalia. This may be an important consideration for all those obtaining copies of documents from the LSS, but this

may not be the best way to present this data. It is recommended that alternate ways of conveying QA Level Code, Media, and Document Condition be investigated.

Based on this survey and on comments from the users, changes to the header design are recommended. These recommendations fall into three groups: fields to be deleted, fields to be revised, and fields to be added.

Many users had difficulty with the location field. The controlled vocabulary for this field is the Places category of the thesaurus and is subject to thesaurus rules and hierarchies. Users made use of the field (in 7% of the queries for a total of 184 occurrences), but two factors contributed to some user frustration. Users had difficulty distinguishing between a geographic location and a named geologic formation, the former being a location term and the latter being a descriptor term. The second factor was that as new locations arose during the cataloging process, they were not automatically added to the location field but were placed in the identifier field pending placement into a hierarchy. This meant that users had to look in the identifier field as well as the location field for a desired term. It is recommended, therefore, that the location field be merged with the descriptor field.

Title and publication data fields need to be re-examined with a view toward moving some of the title information currently contained in the Publication Data field into the Title field (see discussion in section 2.2.2). It is recommended that there be a "macro" capability for searching Title and Publication Data together or separately. (The same capability is needed for Descriptors and Identifiers.)

Certain fields presented difficulties for many users (Document Type, Detailed Document Type, LSS Pointer). Users did not want to eliminate these fields, merely wanted them to be made more understandable. Users had particular trouble with the distinction between Reports and Publications and between Reports and Governing Documents. A clearer structure is definitely needed for document type and detailed document type, particularly if the data base is partitioned by document type. Users do not want to guess where to find a particular type of document. (See also discussion in section 2.2.3.) The concept of the pointer field was understood by users, but the implementation was not clear. Users thought pointers should be bi-directional and include at least part of the title of the pointed-to document. It is recommended that these three fields be reviewed for implementation of a more useful structure.

Additions users would make to the header varied greatly. Users wished for additional fields or additional information in existing fields (event date, enclosures, major subject field, meeting attendees, maps). It is recommended that an event date field be added to the header (see discussion in section 2.2.4). Enclosures are contained in the pointer field (which will be revised to make this more apparent). Meeting attendees, maps, and major subject should

be further investigated for their usefulness as a separate field or in combination with another field.

Users also requested fields that would include full text information from the document such as a table of contents, references or bibliographies, index, list of figures and tables. Users also wanted to specifically search such portions of a document in order to comprehend the content and usefulness of the document. If an unpartitioned header data base is available to users for searching, displaying this information independent of document text would aid in identifying documents containing the desired information without placing a load on the full text system. It is recommended that these areas be investigated to determine the effectiveness and impact of implementing them as header fields.

4.2.1.2 Descriptors

Subject cataloging was captured by three fields: descriptors, identifiers, and location. Assigning descriptors to a document is a very judgmental operation. How well it is accomplished depends on the person doing the assignment, as two people looking at the same document might assign different descriptors depending upon their subject bias. The ideal person is someone who has experience both in a particular discipline and in information storage and retrieval and also understands the licensing process. The catalogers for the prototype were given strict rules for descriptor assignment based both on the content of the material and on how they anticipated it would be used in the licensing process. The latter is very difficult since issues may arise that are not apparent at this time. In general, the Regulatory and Licensing Group did not trust the descriptor assignment but the Technical and Engineering Group did. This may be due to experience on other systems as well as to the nature of their work environments.

On the local system, users had the option of specifying major or minor for descriptors, identifiers, and location. This structure was solely for the purpose of testing the depth of descriptors that would best help users and will not be in the LSS. Based on the test results, it appears that users limited their queries to the major category approximately 70% of the time. However, several queries contained terms where both major and minor were combined for a term search (e.g., Faults[MAJD,MIND]), indicating a desire to cover all contingencies. It is recommended that the extent to which descriptors are assigned be re-evaluated prior to capture station procedures simulation, using 30,000 pages.

A mechanism that would be useful to both catalogers and users is the capability of displaying online the structure of a descriptor (broader terms, narrower terms, related terms, scope notes, and perhaps definitions). A possibility for this is to have a function key that would expand a descriptor from the picklist or controlled vocabulary window. The descriptor field was the

most frequently used of all fields and its use nearly equalled that of full text. Users tended not to use the hard copy of the thesaurus, generally because it was buried under question sheets and working papers and some users had difficulty finding their terms in the hierarchy. A more efficient means would be to provide a user with a hard copy of the shorter keyword-in-context (KWIC) list and the capability of expanding the term online. Such a tool would encourage descriptor use and give users full use of this powerful field. It is recommended that the feasibility of this feature be investigated.

4.2.1.3 Special Tools

All users, regardless of experience, would benefit from simple definitions or explanations of the header fields used. Many users requested that these definitions be available online, perhaps in a pop-up screen. Indeed, online, in context help was requested for all areas of the LSS. A suggestion was also made for an online dictionary of organizations to present the structure of the organization and all its pieces/divisions so that an organization can be searched in part or as a whole. It is recommended that online help and definitions be made available to the user. The online dictionary of organizations is not recommended due to the changing nature of government organizations (users would not know what structure existed when the document was created) and the cost of maintaining such a dictionary. "See also" references in the organization controlled vocabulary that point to previous names of an organization (e.g., DOE, see also ERDA, FEA, AEC) would be less costly to implement and would probably be more useful to users. It is recommended that adding these references be investigated.

Another problem for users was the hard copy thesaurus presentation. One user suggested a permuted term thesaurus, others wanted a KWIC presentation. Users also wanted descriptor definitions expanded; scope notes were not clearly understood by all users. The mechanism for producing hard copy listings other than a hierarchical listing was not in place for the prototype system. It is now possible to produce alphabetical, hierarchical, and KWIC listings in hard copy through the LEXICO thesaurus software. Users also valued the concept of "exploding" a descriptor during a search, i.e., having the option of retrieving all documents where a term narrower than the query term has been assigned. It is recommended that such a feature be made available.

4.2.1.4 Search Strategies and Search Aids

Users of the LSS fall into three categories: the occasional or inexperienced user, the intermittent user, and the heavy or experienced user. The LSS must be hospitable to all categories of users. Generally, this means an easy-to-use, comprehensible user interface for the occasional, inexperienced, or intermittent user, and a command mode with means of bypassing user aids for

the experienced user. How the system encourages or discourages effective search strategies through the absence or presence of searching aids concerns all types of users.

Search strategies using the header included field searching, Boolean combinations of fields, and full-text searches of fields. The fields that users ranked as most or least valuable for searching were shown in Table 4-2. Based on examination of the fields that were actually used during the prototype test (Table 4-3), one can see that the same fields appear, in a slightly different order.

TABLE 4-3
FIELD USE (ACTUAL)

Most Used		Least Used	
Descriptors	18%	QA Level Code	0%
Author Organization	15%	Page Count	0%
Title	12%	Document Condition	0%
Date	9%	Media	0%
Detailed Document Type	8%	Special Class	0%
Location	7%		
LSS Accession Number	6%		
Author	5%		
Abstract	5%		

Figures are percentages of queries in which field was used.

Notes: All other most used fields were used less than 5%.
All other least used field had at least one occurrence.

The use of fields compared with the use of full text in searching is discussed in section 4.2.2.

Selecting a field is the first step in a header search. Since many of the users will be using an assisted as opposed to a command mode, a better presentation of the header field list (the prototype listed them as they appeared in the record) is essential. Suggestions for display included listing fields alphabetically or in order of common use and having the entire list appear on one screen even if this meant using two columns. Users also wanted an easier means of selecting a field from the list, e.g., selecting by number or typing in the first few letters. It is recommended that the header field picklist be reordered and displayed on one screen and that a mechanism for easier selection from a picklist be determined and implemented.

Users on the remote system had the advantage of picklists showing all terms that were used in that partition of the data base with the number of occurrences of each term. Again, users desired an easier means of selecting from the picklist. Some users thought that all terms for a field should be in a picklist for that field regardless of whether the term was in that partition. Another suggestion along this line was to have a "note directing" users to other partitions containing occurrences of the same term. Still another option along this line is to have an unpartitioned header data base for searching. Once users determined the documents they wished to view, they could access the full text through the appropriate partition. It is recommended that the concept of an unpartitioned header data base be investigated with a view to implementation.

In several instances, users wanted the capability of searching two fields as one. These fields included title and publication data, descriptors and identifiers, location and identifiers. Of course, an easy way of doing this is to merely specify the term for each field. However, the idea behind this request is that the occasional or intermittent user would not remember that another field might contain similar data. The concept users had in mind would be similar to having, for example, broad title and narrow title. Broad title would contain both title and publication data fields and narrow title would contain only the title field. Users would like the capability of toggling between the two types at any time during the search. It is recommended that such a feature be adopted.

A major barrier to effective search strategies was the uncertainty users felt in effectively searching for header information. Users want to be able to header search a word or phrase, either field-specific or across fields. Picklists were extremely useful for users on the remote system. They prevented searching for terms that do not appear in a field, provided an indication of how many hits a term has, and allowed users to see "near spellings". (It should be recognized, however, that more experienced users may prefer to use a type-in mode and skip over the picklists.) Picklists for fields containing controlled vocabularies contained the controlled vocabulary terms regardless of whether the term was composed of one or several words. Text fields (such as title, abstract, publication data, and comments) also had picklists, but these contained only single words. Some users had difficulty in understanding how to search for phrases in these fields, particularly those who did not understand that a field was a context unit. Users requested full-text features for searching on any header field that does not have a controlled vocabulary. Users also wanted search terms in header fields highlighted. It is recommended that this barrier be removed by a combination of features: full-text search and proximity searching within text fields, in context online help, and user training.

4.2.2 Full Text

The intent of the user test was not to determine the usefulness of full-text searching but to understand how users use full text for information retrieval. Specifically, what search and retrieval strategies will users employ and how does the system encourage or discourage effective user strategies.

Users searching the remote system had to specify a field for each term entered. "Text" was the field that allowed a user to search the ASCII text of a document. The local system allowed a user to enter a search term without specifying where the term was to be found (either a particular field or document text). Therefore, the analysis of full text in queries is limited to remote system users. These users answered a total of 333 questions with an average of 6.77 queries per question. In 478 or 21% of the queries, users specified the Text field. The breakdown by usage type appears in Table 4-4.

TABLE 4-4

FULL TEXT USE

Type of User	# of Occurrences	% of Use in Queries
Intermediary	34	11%
Management & Administrative	134	47%
Public & Public Information	8	6%
Regulatory and Licensing	172	25%
Technical and Engineering	204	24%
Total / Average	552	24%

Based on these statistics, management and administrative users appear twice as likely and the public information users the least likely to use full text strategies compared to other groups. On the other hand, if the header field use of these same users is compared to their full-text use, the results show that public information users have the highest header field use, but intermediaries have the lowest (see Table 4-5).

TABLE 4-5

FIELD AND FULL TEXT USE

Type of User	% of Queries Using Full Text	% of Queries Using Fields
Intermediary	11%	61%
Management & Administrative	48%	78%
Public & Public Information	6%	98%
Regulatory and Licensing	25%	68%
Technical and Engineering	24%	85%
Average	24%	76%

Note: Full-text and field use will not equal 100% since some queries were combinations of prior result sets and some used combinations of full text and fields (as well as prior result sets).

The table also shows that users most likely to use a combination of fields and full text are the Management and Administrative Group followed by the Technical and Engineering and Regulatory and Licensing Groups, the latter two comprising the largest group of users. It was anticipated that users would use a combination of descriptors and full-text for subject searching, perhaps using descriptors to obtain a working set of documents and then narrowing the search with full text. Sampling the data from questions where users could use this approach showed that this was not the case. Users appeared to be either full-text searchers or descriptor searchers, switching approaches if one did not seem to work, but not combining them. It is recommended that effective strategies for all types of queries be fully covered in user training.

Users were questioned on their use of full text for two specific instances: searching the contents of a table or chart and searching for chemical, scientific, or mathematic formulas or symbols. No user expressed a need to search for formulas or symbols, and only about one-third of the Technical and Engineering groups indicated any need to search the table or chart contents. Most users said they would rely on the image for this type of data. This has major implications for ASCII conversion and formatting, especially since the technical and engineering users are expected to be the largest group of users. (Text format issues are discussed in section 2.3.3.4.)

Picklists were available to users of the remote system and were found to be very helpful. However, phrases did not appear in the picklist for full-text, and users wanted to be able to select combined terms, such as waste package, Yucca Mountain, site characterization. Users on the local system appeared to have greater success since they could search for waste adjacent to package. In the remote system, the smallest context unit was a sentence and the closest

users could get was waste in the same sentence as package. This could lead to false hits in the case of sentences such as "The money spent on sending this package of documents was a waste of funds." For this reason, adjacency searching was thought to be an absolute must for full-text searches. Users specifically requested: term "adjacent to" term, and term "within _ words of" term. Users also desired to use the "wild card" feature of the type-in mode with a picklist to pick up all variations of a term. Adjacency searching and wild-card features are recommended for this system.

Users also had difficulty with the concept of context units as applied to tables of contents, indexes, references, etc. On the remote system, a table of contents was one context unit. Users searching for a phrase in this case might find the two words of the phrase separated by many pages. It is recommended that shorter context units be available for these areas.

One of the major problems of full text is that a concept may occur under many synonyms (e.g., explosions, blasts, detonations, etc.) or that a term may have many forms (e.g., drill hole, drillhole, drill-hole). Users recognized this, and some of the features they mentioned were ways to solve this problem. These features included such things as synonym searching, near spell feature, hashing mechanisms, term linkages, etc. It is recommended that this class of features be investigated for feasibility in the LSS.

4.2.3 Searching, Retrieval, and Post-Processing of Results

Although the prototype data base is a relatively small data set compared to the LSS, it is large enough to give all users an appreciation of the problems of retrieving desired information from a sizeable data base. Indeed, several users who originally did not see the necessity of having a header came away from the test with an appreciation of the use of a header in a data base the size of the LSS. It is clear that certain fields (e.g., title, author, author organization) are absolutely essential for retrieving specific documents, while other fields (e.g., abstract, publication data) are needed for determining the usefulness of retrieved documents. The user who wishes to retrieve all instances of a particular term or set of terms will undoubtedly use full text. The following section discusses LSS Prototype Test Plan questions dealing with special tools for aiding searching and retrieval, strategies users will employ, how the system encourages/discourages effective strategies, and the need to post-process results.

4.2.3.1 Special Tools

In general, while users were satisfied with the makeup of the data base (headers, text, and images), users expect the LSS to have user interface features comparable to existing software packages such as word processors, data base

software, etc. (e.g., scrolling, pop up or pull-down menus, word searches in a document). While most users would be satisfied with state-of-the-art features, others desired artificial intelligence to assist them in finding information.

One of the most important features for searching is an easy mechanism for selecting and combining terms from either fields, full text, or both. Users should have the option for selecting a field from a picklist or of typing in the field acronym. Once a field is selected, the controlled vocabulary should immediately appear. Users who know what term they wish to search should have the capability of typing in the first few letters and have the anticipated term highlighted for selection. Users should be able to easily select more than one term. Users should have the option of combining fields and text, but "text" should not be a field. (This was especially confusing to users upon display when the header appeared as the first "page" of the document.)

Users should have the option of using the Boolean operators between terms or should be asked whether the result needs to meet all the criteria or any of the criteria. Another option would be to have a system default of "AND" between terms. Users should be shown the query as it is being built and in a manner that will indicate how it will be executed (probably through the use of parentheses). They should have the option of editing or canceling a query before it is executed. The option of canceling a query during execution should also be available. Users should be provided with a "working" or "processing" indicator during search execution to let them know the system is still up and running. Strategies that retrieve large sets should display to the user a message, such as "1,000 hits retrieved, stop or continue".

Once the users are ready to view the result set, the option of defining their own display should be available. This user-defined display should be available during the entire session, but users should be able to edit or redefine the format at any time. A short format option should be available and should scroll on the screen, with more than one hit listed per screen. Switching display modes should be an easy operation.

Although many users believed that full-text searching was the best way to find information they knew existed, they appeared to have difficulty managing the text that was retrieved in a hit despite the unitization of documents into smaller units. Some users expressed frustration at not being able to find the information they knew was in a particular document without paging through the entire document. This led to several suggestions by users to assist in finding this information quickly and easily (e.g., searching for terms within a document). Users had a very low tolerance for the number of keystrokes it took to perform an action.

Users should be able to display a reasonable amount of text around a hit rather than just a few words on either side. The amount of text (perhaps a paragraph or two on either side of the hit) should be enough to give the user

a sense of the context of the hit. Users then wanted to be able to move from seeing a hit in context to seeing the full text of the part of the document in which the hit occurred.

Users also expect the capability that exists in word processors of searching for terms within a document but without affecting the query that retrieved the document. For example, a user looking for information on faulting has retrieved a set that appears to give him the information. Within a lengthy document, he may wish to find the section discussing a particular fault, yet without changing his original query or result set. This feature would enable him to search the displayed document for the pertinent section and then proceed to the next document in his result set. In another instance, a user may have retrieved a set of documents using a descriptor. One of the documents may have used a synonym for the descriptor. The user may wish to search the document for all instances where the synonymous term is mentioned. This feature would eliminate the need for the user to return to the search mode, modify his set, and retrieve a new result set, and begin browsing through the documents anew. It is recommended that this feature be made available.

Only the terms in the query should be highlighted, not the entire context unit. This was especially important for tables of contents, references, and indexes. Users had great difficulty finding the search term in these areas.

A way for moving from a hit in a table of contents, list of figures/tables, or the index to the relevant page should be investigated for inclusion in the system. It is anticipated that providing such a mechanism from a reference contained in the document to the document that is the reference might not provide results commensurate with its cost. Consideration of this feature should be evaluated in these terms.

Users should also have the capability of retrieving pointed-to documents directly from the LSS Pointer Field. Ideally, users would like to have a header displayed, place the cursor on the accession number of a pointed-to document and have its header displayed, perhaps in a window or on a split screen. After examining the document, the user should be returned to the original header to repeat the operation if he desires.

The status box needs to be redefined and users should be able to toggle this display on and off as needed with the exception of "document # of X documents"; users wished to have the full screen available for text display. Only those terms in the search query should be highlighted in the text or header field, not the entire context unit. Users should have the option of viewing the hit in context, context in this case being one or two paragraphs on either side of the hit. The ability to toggle between a hit in context and in full text should be available.

4.2.3.2 Strategies

The combination of header and full-text searching is required in the LSS and many users intend to take advantage of its benefits. The extent to which they do so depends on the solutions to three problems: how well the system will handle the terms in a query (variant spellings, phrase searching, wild cards), how quickly a user can find the information retrieved (hits in context, searching within a retrieved document, navigation tools such as go to page #, first page, last page), and how easily sets can be narrowed for display (hits ranked by relevance, marking desired hits for display, display of number of times query term appears in each hit). It must be remembered that the test users were operating on a small body of knowledge (2000+ documents) which is only a fraction of the documents that will be contained in the LSS. The LSS needs to contain features that will enable users to make full use of header and full-text searching without undue frustration. Such features should include mechanisms to assist the user in forming a query, in selecting hits to display, and in finding the information in the hits.

Features that are recommended to assist users in these areas include the capability of editing queries, canceling queries (either before or during execution), and saving queries for use at a later date or for execution across partitions. The system should also display the query as it will be executed. At the time of constructing a query, all executed queries should be displayed with their result sets, including sets with zero results. To decrease user frustration and increase search effectiveness, a maximum query length should be set with a limit to the number of operators in a query. It is recommended that these limits be determined and set.

Users seemed overwhelmed with result sets larger than 6 to 12 documents and started narrowing sets without viewing any of the documents. The low tolerance for the number of documents to be displayed may be due in part to the test itself, not individuals or personalities. Many users seemed to want to answer the question and go on, rather than work through a complex search and large number of documents. LSS usage patterns may vary greatly with some users having more patience than others with large sets. In any case, result set size should be able to be reduced by the searcher. Suggested ways of doing this include:

- 1) Providing some means of ranking documents in a set by relevance.
- 2) Allowing users to mark from a short form display only those records to be viewed.
- 3) Displaying the number of hits of a search term within a document, on a document-by-document basis.

4.2.3.3 Conclusions and Recommendations

Detailed recommendations on specific header design, full-text features, and search and retrieval features appear above in each section addressing specific user test issues. Important general conclusions are:

- 1) System must have ease of use, clarity of choices, help (online or through an intermediary), and must support experienced and occasional users.
- 2) A balanced header/full text search approach is essential; no one feature will appeal to such a diverse audience.
- 3) A variety of strategies need to be supported to master such a large data base with such a variety of needs.
- 4) Limits (query length, hit size) and indicators (processing, percent searched) are needed to encourage effective strategies, build confidence, and enhance interaction.
- 5) Cataloging by experienced (information systems orientation) and knowledgeable (subject experience and licensing orientation) is needed to build user confidence in the system.

4.2.4 Images

The Licensing Support System Conceptual Design Analysis (DOE, 1988) documented the necessity of providing electronic images within the LSS to support the printing of document hard-copy for the users. Although images would be available within the system, the question remained as to how and if the images would be available on-line to the user. To provide answers to these questions, the prototype system provided two basic types of data. Two image workstations were provided with images for a portion of the database available on optical disks local to the workstations. Manual counters were located at the remaining four workstations which were incremented whenever a user would have liked to view an image on-line. The image workstations provided qualitative data relative to the functional design of the LSS image workstation, primarily through comments elicited from the users and observations by test personnel. Unfortunately, the software which provided the images could not readily be instrumented to provide a count of images requested at those workstations. Quantitative data was available only from the counters in the non-image workstations.

The thrust of the test plan on the subject of images was to determine how often users would request images and how useful are images of non-textual pages. Even though the question of images was not a major objective of the prototype

tests, useful information was obtained which will be used in the determination of system requirements.

Results from the tests fell into two major categories as discussed below. The first category pertained to the general usefulness of on-line images, particularly as they related to the set of questions represented in the tests, and a measure of the frequency of use. The second category described certain functional capabilities which should be available in the utilization and display of images.

4.2.4.1 Usefulness of Bit-Mapped Images in the LSS

Twenty-three users were assigned to workstations without image capability during the prototype tests. The number of requests for images (if they had been available) during the seven-hour test period ranged from zero to 23 per user with an average of 3.1. Figure 4- 1 shows a frequency distribution of these requests. The two usage groups which exhibited the highest average requests per user were the Technical & Engineering group and the Regulatory & Licensing group.

Observations on the usefulness of images in the prototype were made primarily by those who used the workstations equipped with images. The comments regarding images in general ranged from "not useful" to "all documents should be imaged". The primary usefulness of images was to view information which is not contained in ASCII form, i.e. maps, graphs, illustrations, etc. This was particularly important to the Technical & Engineering usage group. Less frequent observations were that images were useful for observing marginalia and date stamps. (The users expressing that opinion were probably not aware that marginalia and date stamps will only reflect the first copy of that document entered into LSS.) The strength of the observations from the prototype users confirmed the conclusions of the LSS Preliminary Needs Analysis (DOE, 1988a) that images are required at least for graphical data which cannot be made available in ASCII form. There is no strong need expressed for images of information available in ASCII form provided that the users can have reasonable assurance of the accuracy of the ASCII and the format is maintained to the extent possible.

The availability of images on-line (as opposed to images only available to produce hard-copy) is a function of two factors. The first is the speed in which the user requires the response, i.e. can the user wait for an overnight mailing? The second aspect is whether or not image information is necessary on-line to aid in determining which documents are relevant to the user's query. Based on observations from the tests, it would appear that on-line images will be very important to the Technical & Engineering usage group in determining document relevance and probably occasionally in speed of response.

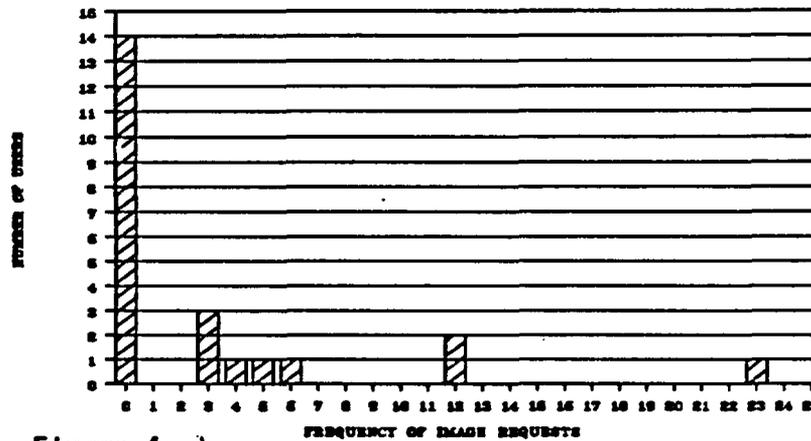


Figure 4- 1.

4.2.4.2 Display Functionality Requirements

Users of the image workstations provided many useful comments on the functional requirements for the future LSS workstations. Of particular importance was that the images provided were judged quite usable at the resolution presented. The images for the prototype were captured at a resolution of 300 dpi x 300 dpi and displayed on a screen at 300 dpi horizontal and 143 dpi vertical resolution. One user commented on the fact that the photographs were not accurately represented, however this would be expected since the monitors did not provide grey-scale capability. Most users also agreed that color would not be worthwhile for capturing or displaying images for the expected limited data in which color differentiation would be important. The screen size used was generally considered adequate (capable of displaying a full 8 1/2 x 11 inch image in portrait format), however a few users expressed the desire to simultaneously display ASCII on a split screen.

Users were similarly in accordance in their comments regarding the difficulties and awkwardness of the user interface in requesting images to be displayed from the ASCII text. (The basic functionality was provided by the Information Dimensions, Inc. GTerm DOS-based image handler.) Of particular note was the difficulty in positioning the cursor or text on the screen to be able to display a particular image. Frequently the image recalled was not the image expected.

A second major problem was the difficulty in "navigating" through tiled images (oversize documents captured in a series of 8 1/2 x 11 inch images). This problem derived from two aspects of the prototype. The first was that no indication was provided to the user as to which tile was being displayed at the moment (unless this could be discerned from a border visible in the image). The second was that it was very difficult to determine which "image marker" in the text represented the desired adjacent image. Zoom and pan functionality was frequently mentioned as a solution to this problem. The system should be smart

enough to automatically pan from one tile to another, while "zooming" out would provide an overall view of the complete image.

The particular software used for displaying images required the ASCII text to be displayed and the images could then be accessed from the text. In the user debriefing, the question was asked about the need for accessing images directly from the header display, without going through the text display. The responses were split with the majority responding that recalling images directly from headers was not necessary, and a few responding that it would only be necessary when the record did not contain text. Only two users mentioned the desirability of paging from image to image without the necessity of going back to the ASCII text to recall the subsequent image.

4.2.4.3 Image Conclusions and Recommendations

Electronic images are required in the LSS to meet the requirements of 10 CFR Part 2 and to support the printing of hard copy. On-line images are sufficiently important to a segment of the LSS user population to justify the inclusion of this functionality in the final system. Indeed, one user expressed the opinion that electronic images was the major distinguishing feature of the LSS and without that function, a combination of existing systems could provide most of the necessary LSS functions.

Capture of images in black and white at a resolution of 300 dpi and display resolution of 300 x 150 dpi meets the expectations of the users for on-line display. The prototype tests did not determine if lower resolutions would also be acceptable. The monitor used in the LSS should, as a minimum, provide a full 8 1/2 x 11 inch page image in portrait format. Simultaneous display of ASCII information would be helpful. If the image is displayed in an overlapping format as in the prototype, the image should be erasable.

Images can be accessed through the ASCII text, however the correlation between the image recalled and the text must be much easier for the user than in the prototype. Paging from image-to-image is a desired function, but interest was not strong enough to warrant making this a mandatory function, except as noted below for tiled images.

Navigation through tiled images requires definitive improvement in the user interface. However, it should be noted that the Capture System Design Document establishes the maximum size of oversize documents to be captured in the LSS at 17 x 22 inches which would only require 4 tiles. This fact in itself will alleviate the problem in getting lost in the interior of a many-tiled document. (The prototype data base included documents containing as many as 36 tiles.) Zoom and pan capability should be provided, including the capability of panning from one tiled image to the adjacent image. Another solution would be to display the four images on the screen in reduced scale and zoom to the desired image.

A simple graphical image of four quadrants indicating the current quadrant being displayed would provide the user with sufficient information to avoid being lost.

No quantitative information was gained from the prototype on the frequency of images requests in order to assist in the design of the communications system or the workstation. General observation was that users did not express a strong desire to "page through" images in succession. Viewing of images was generally associated with graphical figures referenced in the text rather than a means of reading the text itself. Therefore in general there was no strong demand for a rapid sequence of images. One exception to this case would be the oversize or foldout figure which must be tiled. Consistent with the functionality described above, the capability of downloading four images in rapid succession and storing them locally at the workstation would be required.

The prototype database contained several instances in which blank pages were scanned and images maintained. As the users found no real usefulness in images of blank pages, it is recommended that the LSS database not contain images of blank pages.

4.2.5 General

This section discusses the results and observations from the prototype tests relating to several issues: Partitions, displays, system response, and printed copy. Specific questions from the test plan which will be addressed include:

Can users work with the proposed partitions (i.e. organized by type and date)?

What displays of post-search processing results are useful (including hardcopy)?

4.2.5.1 Partitions

This issue elicited probably the greatest range of responses and the most negative comments of the tests, even though the partitions were only implemented on the remote system (four of the six user terminals). The implementation was particularly user unfriendly, requiring users to reenter the query for each partition. This was done in order to determine the minimum acceptable implementation for the LSS. Assuming that partitions will be required for the LSS, two aspects require particular attention. The first relates to how queries are performed on the partitioned data base, and the second relates to the methodology for partitioning the documents.

Comments from the users were almost universal in agreement that the implementation utilized in the user tests was unacceptable. Typical comments included phrases like: not user friendly, too many steps to change partitions, tempted to trash the system, cumbersome, too much burden on user, least desirable feature of the system, and very dissatisfied. At the very minimum, a capability to save a query and apply it to another partition is required. A preferred concept would be to apply a given query to selected partitions (including all), and return a matrix of hits versus partitions. The user could then continue queries in partitions which yielded the most promising results. A refinement on this concept would be to provide a "partition-less" environment for queries on headers, but implement partitions for full-text queries.

Other than the inability to query across partitions, the second greatest problem was understanding the definitions of the partitions. The test data base was partitioned by document type, with the reports being further subdivided by date. The greatest difficulty was to understand the document type definitions. Particular areas of confusion involved reports vs. correspondence, governing documents vs. reports, publications vs. reports, and reports vs. data. On-line help in understanding the definition of partitions (document types) was mentioned as a partial aid in alleviating the situation. Various suggestions were received on how to best partition the data base, and they have been categorized below. (The number in parenthesis is the number of users who expressed that or a similar view.)

- Partition by date (9)
- Partition by document type (6)
- Don't partition by date, too many different dates involved (6)
- Don't partition by document type, subject to judgement (2)
- Partition by subject is a poor idea (2)
- Partition by publication date (1)
- Partition by agency is a poor idea (1)
- Partition by document size (1)
- Partition by statute (1)
- Partition by subject (1)
- Partition by Adjudicatory Proceedings (1)
- Partition by technical discipline (1)
- Partition reports by pre- or post EA (1)

Despite the comments, the statistics do not appear to indicate that the partitions were an overwhelming problem to the user, even in the prototype implementation. Since the system initialized into the Correspondence partition, it can be expected that the number of partition changes for a given question would frequently be at least one, and indeed this was true for 97% of the questions. It is interesting to note, however, that 75% of the questions required 3 or fewer changes of partitions. (The statistics do not indicate whether or not the questions were successfully answered.) Some persistent users

tried as many as 19 partition changes per question. See Appendix A, Figures A-6 through A-8 for details on partition changes.

Several users did express positive comments about the partitions with three indicating that partitions caused no real problems. One indicated that partitions excluded irrelevant material and therefore speeded up searches, and two commented that partitions forced the user to think about the strategy before proceeding.

4.2.5.2 General Display Issues

Comments with respect to the screen displays included likes, dislikes, and suggestions for additions. Most of the comments received pertained to the remote system. Therefore all comments can be assumed to come from users of that system unless noted by the "(local)" after the comment. Display comments include:

The function key display was useful. One user suggested a one-line display would have been preferable.

The result set display was O.K., but should have been more prominently displayed.

The display of LSS accession number and document title in the upper "banner" was useful.

Starting a display at the top of the document is preferable. The header should be separated from the document and not displayed as the first page.

The number of keystrokes required to accomplish a task (i.e. changing format of the display) was excessive.

The ASCII text display was too small, causing users to browse images instead. Provide the ability to get rid of the boxes and expand the screen.

There was a general inability to move freely within the document. The same capabilities as a word processor software are expected.
(local)

The dashes outlining the boxes was annoying. Some users disliked the amber/black display.

Desirable features include:

Ability to rank documents in displaying "hits" by order of importance

Ability to customize header format being displayed

Easy access to table of contents

Ability to go to specific page in the text

Ability to advance or return "n" screens

Windowing environment

Ability to view previous result set while formulating query

Single keystroke to change display formats

4.2.5.3 System Resources

There were few complaints with regard to the response time of the systems in response to a query. Most users commented that it was acceptable or good. This applied to both the remote and local implementations. The response time of the remote system to a query was one second or less for 95% of the queries. The response time of the local system was significantly slower, with 95% of the queries returned in 45 seconds or less (See Table B-4). One aspect of the remote system was noted as being particularly slow, however, and that was the response to a request to display the next occurrence of a full-text hit in context. This aspect was particularly annoying in light of the absence of a "system working" indicator.

The times for the retrieval systems to return the answers to the formulated queries is the best estimate of the amount of CPU and I/O time required. For both systems, times were measured to the nearest second. Since all remote system were one second or less, no good distribution of the effect of query complexity on response time could be obtained. For the local system there was a much wider range of response times, ranging from 0 to 144 seconds. The distribution of response times as noted in Appendix B, Data Set B-7, has a very long "tail" of values greater than the mode (most common value). The mean value is 11.8 seconds which is nearly seven times the mode of 1.7. Ten percent of the responses required nearly eighteen times the resources of the most common query. Thus some queries have the potential of generating large resource demands.

Review of the queries associated with the longest response times indicated that the common factors were the use of wildcards and context operators. However, some queries associated with shorter response times also used wildcards

and context operators. A detailed analysis and study is required to develop a clear relationship between query complexity and response time.

In response to a question asked during the exit interviews, "acceptable" maximum response times to queries varied widely, from 10 seconds to 1-2 minutes (the latter for full-text searches). One user suggested that the ability to batch queries for overnight processing would be useful.

Users tended to want to browse anywhere from 10 to 50 pages of text at a time. Actual responses were as follows:

<u>No. Pages</u>	<u>No. Responding</u>
5-10	1
10	3
15-20	1
50	4

Section 4.2.5 profiles a "typical" prototype user based on the statistical results from the tests. This information can be used to determine the frequency and severity of demands on system resources for the purpose of determining system requirements.

4.2.5.4 Need for Printed Copy

There were no downloading or print functions associated with the prototype user tests, so no quantitative data is available to assist in defining the required functionality. However, users were asked for their thoughts and preferences on this subject during the exit interviews. Users defined three functional areas which will be important to future LSS users.

Many users indicated that their common mode of operation will be to order hard copies of documents to be studied in more detail later.

Technical users in particular desire the capability to download text, parts of documents, and images to a PC. Intermediaries wanted to have the capability of downloading result sets and headers to PCs for sorting and editing.

Several users expressed the desire to print locally for screen dumps, headers, etc.

4.2.5.5 Conclusions and Recommendations

With respect to the partitions issue, it is clear that a "friendlier" user interface is required. The ability to save queries and apply them to different partitions (with perhaps an edit capability) would be a minimum requirement of the system. The ability to simultaneously parse queries to several partitions and report the number of hits by partition is a preferable alternative.

The methodology of partitioning the data base is not as clear. It appears that date would be the preferred alternative since it is important to choose a discriminator which can clearly be defined and includes a minimum of subjectivity. Particular dates chosen for breakpoints would better be defined from program major milestones than calendar years.

Designing a user interface which is pleasing to everyone is a difficult task. It is also possible that the user interface for the LSS will not be developed specifically for that application, but may be one that is available on the text DBMS which is chosen for use. It is clear that many of the potential LSS users, particularly the technical and legal types, are experienced personal computer users and will expect a level of sophistication in the user interface equivalent to popular software packages. One overall recommendation from the user tests is clear: Care must be taken to insure that the number of keystrokes to accomplish a given task be minimized.

Response to a query should be provided to the user in less than 10 seconds for most queries. A system working indicator, preferably displaying progress, should be provided.

The typical user will utilize the LSS to find documents of interest and request that copies of those documents be sent to the user for further study. Many users will also require a downloading capability for ASCII files, and in addition technical users would like to download and locally print images.

4.2.6 User Profile

For the purpose of system design, particularly in the areas of computer resources and communications, it is useful to try to distill the data from the user tests into a "typical" user scenario. From this scenario and an assumed system architecture, the demand on CPU resources and communications load can be calculated. For this purpose a user "session" is assumed to be equivalent to the actions taken by the user to answer a single question posed in the test set, and includes logon, query formulation, system response, display of results, and logoff.

This section provides insight into the test plan questions related to resource requirements and system capacity, in particular, how often users perform resource intensive operations.

The average time spent answering a question within the sets provided was 28 minutes. Since these questions were chosen to be representative of the types of questions that LSS users will be answering in actual use, the time period is considered to be valid. The average time for the remote system exceeded the average time for the local system by over 8 minutes, however there are many factors contributing to this differential. Among them are the image capability on the remote system, the unequal distribution of user types, and the fact that the remote system data base was partitioned. Response time, while being a minor contributor to overall session time, was actually much shorter on the remote system.

Queries

During the scenario, the user will issue between 6 and 7 queries to the data base (The average for the tests was 6.3). There were actually considerably more queries initiated, particularly on the remote system. For that design, one of three possible actions were possible once a query was initiated:

- 1). Evidence was available from the pick lists that the query was not going to be useful and it was abandoned.
- 2). An error was made in forming the query and it was abandoned (since query editing was not available).
- 3). The query was executed.

The information available indicates that one query was abandoned for approximately every two queries executed on the remote system. Depending on the system design, the abandoned queries may or may not impose a demand on the system data base resources.

Variation on the number of queries per question among user groups ranged from 5.4 for the IN group to 7.9 for the PI group (See Table 4-1).

The time to formulate and execute a query was on the order of one minute each. For the remote system the average time to build a query from the pick lists was about 80 seconds while the typed queries were developed in about one half of that time. By comparison the query development time for the local system, which only executed in "type-in" mode was about 60 seconds. This is expected since only the more experienced users tended to use the type-in mode on the remote system while both experienced and inexperienced personnel were users of the local system.

One of the major determinates of resource requirements relates to query complexity, i.e. how many fields, terms, Boullian operators, etc. are included in each query.

To determine the Search System computer resource size, a query load analysis should be performed using the distribution of response times available from the local system, scaled to the specific hardware. The scale factor would be established by running a sample set of queries against the prototype data base on the computer in question.

Display of Documents

A second determinate of system resource requirements is dependent upon the actions taken by the user in displaying documents on the screen. Typical activities included displaying headers, displaying full text, and in limited cases, displaying images. The discussion relating to images is covered in Section 4.2.3.

It is somewhat difficult from the available data to determine the number of times that a user displayed documents on the screen, however the indications are that on the average, a user looked at documents about 3 times during a session, or once for every 2 queries. Each display period lasted 4 to 5 minutes on the average, however some users spent as long as 45 minutes in a single display period.

Use of functions to navigate through documents was recorded only for the remote system (Reference Appendix A, Data Set A-2). Based on these figures, users are expected to perform on the average about 12 next-page or next-screen commands per scenario and 12 next-document commands. Arrow keys could also be used to move through text, however these were not recorded. In addition users navigated backwards through the text with previous-page and previous-screen commands, however these were used much less frequently, averaging only twice per question. Moving backwards through documents with the previous-document command was also less frequent, also averaging about twice per question. Variation on use of these functions by type of user was not highly significant, but the TE user group tended to use the functions the most and the IN group the least.

5.0 CAPTURE SYSTEM PROCEDURES DEVELOPMENT INPUT

Many of the lessons learned from the prototype testing, particularly in the area of data base preparation, will affect the direction and development of the procedures to operate the Capture Stations. This section is a collection of the recommendations and conclusions which pertain to procedures.

5.1 Initial Processing

Initial processing includes the checking, unitization, accession number assignment, page identification, partial header preparation, and duplicate checking of the documents to be processed for inclusion in the LSS. Procedures for initial processing of documents should be developed along the following lines based on lessons learned from the test data preparation.

The cost of preparing a document for input, particularly the extraction of ASCII text, is highly dependent upon the quality of the submitted material. Costs spent in locating the "best" copy are still less than the costs associated with trying to process poor copies. Therefore procedures for acceptance of documents for processing, as well as procedures for submittal, should impose the burden on the organization submitting the document to track down the "best" possible copy. Documents which are largely illegible should be imaged only as the cost of ASCII conversion and the quality of the result is not worth the effort.

Unitization of documents should emphasize dividing submittals into the smallest identifiable unit. Processing these units individually, while increasing the cataloging costs, is still less costly than duplicating documents which are, for example, attached to several different pieces of correspondence. The emphasis is to reduce the number of pages processed through ASCII conversion and imaging to a minimum. Associated with the unitization process is the development of a quality duplicate check process.

Processing blank pages in the documents is an unnecessary expense and should be avoided.

5.2 Cataloging

Cataloging includes the completion of the header information, including the assignment of subjective descriptors and identifiers.

Titles should be entered into the title field as they appear on the document with preceding articles. In the event that sorting is required, the software should ignore the articles and sort on the first significant word. When

the document being cataloged is a portion of a larger volume (i.e. a chapter of the SCP), the title field should contain the title of the entire volume.

The document type descriptions need to be revisited with the goal of defining less ambiguous definitions. This is particularly important if the document type is to be used as a discriminator for partitions.

The level of definition of descriptor assignments in the prototype development was too detailed. Cataloging procedures need to reflect the assignment of descriptors for major concepts only. The location field yielded little value in document searching and should be folded into the descriptor or identifier fields.

Utilization of acronyms for the "authority list" for organizations is confusing due to potential duplication. Acronyms should be avoided or modified to eliminate ambiguity.

5.3 ASCII Conversion and Image Preparation

ASCII conversion includes the preparation of ASCII text from hard copy and electronic text document submittals. The experiences of the prototype test was limited to the processing of hard copy documents by optical character recognition (OCR) with manual correction or rekeying. Image preparation includes the creation of bit-mapped images from paper or microform documents. The images are used both for storage and display within the system as well as input to the OCR process. The prototype test data base preparation included imaging of hard copy only. Development of procedures for ASCII conversion and imaging should be developed along the following guidelines.

The costs associated with these processes are dependent upon decisions and procedures on the manner in which the documents are to be processed. Unfortunately the multitude of different circumstances which confront the capture process precludes the development of detailed procedures which can be followed by unskilled personnel. Therefore the processing of documents through these steps should be preceded by a preprocessing analysis performed by highly trained staff. Correct decisions at this step on OCR versus rekeying and zoning of images, for example, can improve overall efficiency and accuracy of operation.

Procedures for the preparation of ASCII text can be developed on the basis that variations in page format from the original is permitted. Preservation of capitalization is preferred and word-wrap should be avoided. Line length is limited to 80 characters maximum.

The process of developing substitution procedures for Greek letters, formulas, super- or sub-scripts, and other non-ASCII characters proved to be difficult due to the variety of cases which must be covered. Clear, concise,

and consistent rules are needed to protect the integrity of and confidence in the data base. Fortunately, the tests revealed that detailed substitutions are not required (provided page images are available to the user), and simple notations indicating missing text are sufficient.

Tiling of large documents resulted in much difficulty in user orientation. The number and types of documents containing this type of information do not appear to be sufficient to warrant the complicated display hardware and software necessary to alleviate the problem. The procedures for processing these documents should be based on handling a maximum size of 17" x 22", which can be handled as four 8 1/2" x 11" tiles. The user will be provided with a source from which copies of larger documents may be obtained.

Correction of errors in text prepared by OCR is most efficiently handled by use of spell checking dictionaries, however this results in correcting errors in the original document as well. The cost of determining if the error is a result of the processing or was in the original document is prohibitive, and retrieval is enhanced when spelling errors are corrected regardless of source. Therefore procedures should reflect correction by means of spell check dictionaries without regard to source. The final accuracy of 99.8% proved to be a reasonable compromise between cost and user confidence in the data base.

The final ASCII text product must not contain tabs. The substitution of spaces to maintain format is required.

The definition of context units for text search queries may require processing of the ASCII to ensure that context units are consistently defined. (For example, each sentence should end with a period or question mark followed by two spaces.) Special treatment may be required for title pages and lists of figures and tables. Users expressed sufficient interest in being able to move to the table of contents, and to ensure that the table of contents is not a context unit in itself, that special processing to identify that section is required.

6.0 SEARCH AND IMAGE SYSTEM DESIGN REQUIREMENTS INPUT

A major purpose for the LSS Prototype user tests was to develop a better understanding of potential LSS user behaviors and needs. This section is a compilation of the lessons learned stated in the form of requirements for the LSS Search and Image System. This section will be one of the major inputs to the LSS Search and Image Design Requirements Document.

6.1 Search System

6.1.1 Query and Retrieval

Query Construction

Four query construction modes are needed to satisfy the full range of users.

- 1) Picklist with the following features:
 - quick movement to a term via typing the first few letters in the term
 - easy selection of multiple terms
 - easy phrase construction
- 2) "Type-In" query construction mode
- 3) Modification of a previous query
- 4) Retrieval and reuse of a previous query, even from an earlier session.

Each query should be shown as it is being constructed using parentheses to indicate nesting.

A query should be able to be applied to:

- 1) Single field
- 2) Multiple fields
- 3) All fields
- 4) Text
- 5) Text and any combination of header fields

Wild cards should be available in all query construction modes.

The use of Boolean operators should be supported. In picklists a total query default choice of "all or any criteria" should be selectable.

Proximity and phrase searching should be available for both header fields and full text.

An easy-to-use way to apply a query to a set of partitions is needed.

Other Capabilities

On-screen selection of an LSS pointer to directly retrieve and display the associated header or document is desired.

Full-text search within a particular document, either one picked from the current result set or the one being viewed, is needed.

Indication as to what the relative response time will be for a given query, as well as to the progress of the search, is needed.

A user should be able to cancel a query both during construction and while waiting for the response.

A capability to save queries for use in the same or future sessions is needed.

To assist with the selection of descriptor search terms, there should be some online display of the thesaurus that would show:

- 1) Broader or narrower terms
- 2) Scope notes
- 3) Related terms

Both near spell and synonym search capability are needed to reduce user frustration.

Punctuation needs to be ignored for searching of report numbers in the header.

6.1.2 Display

Layout

The following general display layout features were identified as being required:

- 1) Display a status box, at the user's option.
- 2) Display a top-of-screen banner showing document title and number in current result set, removable at the user's option.
- 3) In header displays, show the number of hits within a document.
- 4) ASCII text display should start with the first page of the document.
- 5) For a large display screen, the following side-by-side combinations should be possible:
 - two headers
 - header and ASCII text - same document
 - two pages of ASCII text - same document
 - two pages of ASCII text - different documents
- 6) Headers and the ASCII text for a document should be clearly separated.

The following display modes were identified as desirable:

- 1) Header
 - one to three lines per document header
 - user-selected set of header fields
 - full header
- 2) ASCII text
 - full page
 - full screen
 - one or two paragraphs before and after the location of a hit

Navigation

Very easy switching between all displays, particularly hits and other ASCII text displays, is required.

The ability to go directly to a page is needed.

- 1) Select a page in a table of contents or list of figures

- 2) Type in a page number

The ability to go forward or backward "n" pages or screens in the ASCII text display is needed.

The capability to mark a document from various header displays, for subsequent viewing of ASCII text, is required.

Other Features

The option to rank hits and display headers in relevant order is desired.

Header displays should be sortable by any set of header fields. The articles (The, A, An) should be removed for sorting titles and organizations.

Only the terms used in the query should be highlighted as hits in both header and ASCII text displays.

6.1.3 User Interface

The user interface design should be based on the following:

- 1) Consistent use of function keys
- 2) Online function key labels
- 3) Single key stroke to switch between query and display
- 4) Clarity of choices presented
- 5) Easy display of the most recent queries and their results in either display or query modes
- 6) Online help features for definition of header fields and explanation of function keys, picklists, and navigation options.

Download capabilities of the following items were identified as desirable:

- 1) Selected ASCII text pages
- 2) Selected headers
- 3) Selected images

Print screen capability and local printing of download items is required.

6.1.4 Resource Requirements

Average query response time should be less than 10 seconds.

System resource demands can be developed from the following user characteristics:

- 1) A typical LSS user session has a duration of 28 minutes and includes the following steps (section 4.2.5):
 - start session
 - construct query
 - construct query
 - view headers and ASCII text
 - construct query
 - construct query
 - view headers, ASCII text and images
 - construct query
 - construct query
 - view headers, ASCII text and images
- 2) The distribution of query resource demands is represented by the DB_RESPONSE times in Data Set B-4, Appendix B.
- 3) The distribution of query times is represented by the BUILD_QUERY and TYPE-IN_QUERY times in Data Set A-9, Appendix A, and QUERY_TIME times in Data Set B-4, Appendix B.
- 4) The distribution of display times is represented by the DISPLAY_DURATION times in Data Set A-9, Appendix A, and DISPLAY_TIME in Data Set B-4, Appendix B.

Loading of the prototype data base revealed that significant resources in processing time and storage will be required, which is dependent upon the selected text DBMS.

The requesting of a document to be printed and delivered, needs be able to be invoked any time during the viewing of documents.

6.2 Image System

Methodology for orientation and navigation through panels of oversize images must be developed.

The following side by side image display requirements were identified

- 1) Header - Image same document
- 2) ASCII text - Image same page
- 3) Image - Image adjacent pages

7.0 REFERENCES

DOE, 1989; Requests for Comment Number DE-RP08-89NV10888 Licensing Support System Category II - Scanning and OCR Data Conversion Resources, DOE Nevada Operations Office, 25 July 1989.

SAIC, 1988; Licensing Support System Prototype Development Plan, Science Applications International Corporation, prepared for DOE Office of Civilian Radioactive Waste Management, 22 January 1988.

SAIC, 1988; Licensing Support System Concept Feasibility Analysis, Science Applications International Corporation, 10 July 1988.

SAIC, 1988; Licensing Support System Prototype Test Plan, Science Applications International Corporation, 26 September 1988.

SAIC, 1989; Licensing Support System Capture System Design Document, Science Applications International Corporation, 9 March 1989.