DOCUMENT RESUME

ED 060 883 LI 003 550

AUTHOR	Epstein, A. H.; And Others
TITLE	Bibliographic Automation of Large Library Operations Using a Time-Sharing System: Phase II, Part 1 (July 1970 - June 1971). Final Report.
INSTITUTION	Stanford Univ., Calif. Libraries.
SPONS AGENCY	Bureau of Libraries and Educational Technology (DHEW/OE), Washington, D. C.
BUREAU NO	BR-7-1145
PUB DATE	Feb 72
GRANT	OEG-0-70-2262
NOTE	287p.;(35 References)
EDRS PRICE	MF-$0.65 HC-$9.87
DESCRIPTORS	Information Processing; *Information Systems; *Library Automation; Library Cooperation; Library Networks; *Library Technical Processes; Man Machine Systems; *Time Sharing; University Libraries
IDENTIFIERS	BALLOTS; *Bibliographic Automation Large Library Operations

ABSTRACT

The main objectives of Project BALLOTS (Bibliographic Automation of Large Library Operations Using a Time-Sharing System) are to control rising technical processing costs and, at the same time, to provide improved levels of service. This report on BALLOTS Phase II is concerned with the development and implementation of the production library automation system - the system that will support the day-to-day operations of the library. The report is divided into four parts. Chapter 1 gives some background for the report and summarizes the nature of the BALLOTS system, as well as its status at the end of the reporting period (June 1971). Chapter 2 describes development progress in two different areas: the bibliographic services and system design as seen by the user; and the software and hardware design to support these services (including video terminal selection and screen design). Chapter 3 describes the major standards and analytic studies completed during the design. Each of these standards or studies became a part of the design, or had a substantial effect on the user, hardware, or software design described in Chapter 2. Chapter 4 describes the activities currently under way (following the reporting period) and future plans. (BALLOTS Phase I Final Report is available as ED 049 786) (Author/SJ)

FINAL REPORT
Project No. 7-1145
Grant No. OEG-0-70-2262

BIBLIOGRAPHIC AUTOMATION OF LARGE LIBRARY OPERATIONS USING A TIME-SHARING SYSTEM: PHASE II, PART 1
(July 1970 - June 1971)

A. H. Epstein, Wayne Davison, Eleanor Montague,
Madeleine Stovel, Allen B. Veaner

Stanford University Libraries
Stanford, California 94305

FEBRUARY 1972

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE

Office of Education
Bureau of Libraries and Educational Technology

The research reported herein was performed pursuant to a contract with the Office of Education, U.S. Department of Health, Education, and Welfare. Contractors undertaking such projects under Government sponsorship are encouraged to express freely their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Office of Education position or policy.

CONTENTS

ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 SCOPE OF REPORT

BALLOTS I was concerned with the conceptual design, implementation, and evaluation of a prototype system of computer support for bibliographic operations in large libraries and with provision for extensions to other libraries. BALLOTS II is concerned with the development and implementation of the production library automation system--the system that will support the day-to-day operations of the library.

The development and implementation of BALLOTS II are divided into six parts. The first of these, General Analysis, was described along with the prototype system in the BALLOTS I Final Report. (The BALLOTS I Final Report covered the period from June 1967 through June 1970.) The next three parts, Detailed Analysis, General Design, and Detailed Design, are the subject of this report, which covers the period from July 1970 through June 1971. Office of Education support for BALLOTS II (grant OEG-0-70-2262) continued through March 31, 1971. In the interest of providing continuity, this document describes work in progress during the grant period but completed during the following ninety days.

The final two portions of BALLOTS II development, implementation (programming, training users, testing) and installation (file conversion, pilot operation, and acceptance testing), are in progress as this report goes to press. At the end of January 1972, programming has progressed to the point where a 1,200-record MARC tape has been converted to an on-line file, full indexes have been built for this file, and on-line searching and display of the search results have been done on both a typewriter terminal and a CRT (cathode ray tube) terminal. The remainder of the acquisition and cataloging CRT screen formats and printed output formats are being coded. Schedules are being drawn up for MARC tapes file conversion and for training computer machine room attendants. The printing forms have been ordered for various outputs, and catalog card cutters are being evaluated. Computer cost accounts are being arranged so that detailed cost figures can be collected for such categories as supplies, terminal rental, on-line activity by library department, printing, file maintenance, etc. Training and reference manuals for users are being drafted. Communication lines are being installed for use with the CRT terminals to be delivered in February and March 1972.

The present report is divided into four parts. Chapter 1 gives some background for the report and summarizes the nature of the BALLOTS system, as well as its status at the end of the reporting period (June 1971). Chapter 2 describes development progress in two different areas: the bibliographic services and system design as seen by the user; and the software and hardware design to support these services (including video terminal selection and screen design). Chapter 3 describes the major standards and analytic studies completed during the design. Each of these standards or studies became a part of the design, or had a substantial effect on the user, hardware, or software design described in Chapter 2. Chapter 4 describes the activities currently under way (following the reporting period) and future plans.

1.2 BACKGROUND

In 1967, Stanford began a continuing library automation project, BALLOTS--Bibliographic Automation of Large Library Operations using a Time-sharing System. A final report on the first phase of BALLOTS (July 1967 through June 1970) was submitted to the Office of Education in April 1971 <1>. That report reflects the work done under grant OEG-1-7-071145-4428. The present report reflects work done under a continuation grant, OEG-0-70-2262. Readers are referred to the earlier report for details of development related and leading to the contents of the present report. However, for the benefit of readers who do not have ready access to the earlier report, a brief summary is given in the following sections, 1.3 through 1.6.

Project BALLOTS is directed toward maximizing the contribution of the large library to university education. (Through its generalized design and network capabilities, BALLOTS facilities are extendable to libraries of various types and sizes--see section 3.7). Increasing costs of operation and the limitations of a manual file system inhibit the library's response to the changing information requirements of higher education. The BALLOTS approach is to provide technological assistance to the library and the academic community in the form of an on-line production bibliographic processing system. The project has been conducted in two major phases: (1) BALLOTS I, research and prototype development, completed at the end of calendar 1969, and (2) BALLOTS II, production system development, the present activity.

BALLOTS has been conducted as the collaborative effort of the Stanford University Libraries, the Institute for Communication Research, and the Stanford Computation Center. The project is monitored by an executive committee chaired by Provost William F. Miller. The BALLOTS project director, Mr. A. H.

Epstein, holds a joint appointment with the Library and the Computation Center. Shared software (see section 1.5) is being developed with the Stanford Institute for Communication Research, which is conducting Project SPIRES--Stanford Public Information REtrieval System. SPIRES is funded by the National Science Foundation. With a project structure that links the campus computer center, related information retrieval activity, and the working library staff, it has been possible to coordinate operating needs, technical factors, and design capabilities.

1.3 THE PROBLEM

The large research library is caught between the spiraling costs of maintaining a crucial but cumbersome system of manual files and the demanding social and educational forces that are reshaping the university environment. Labor costs have mounted sharply while large libraries have had to enlarge their staffs merely to cope with the increased size and complexity of their files. Despite current cutbacks and leveling off in the rate of library growth, there is good reason to believe that the overall trends will be upward to the end of this century. In this same period of rising costs, information and educational technology have made great strides. This is indicated by the availability of machine-readable data bases (such as MARC tapes and census tapes), computer-assisted instruction, the wide use of audiovisual equipment, the continuing experimentation with cable TV, and the development of regional computing networks.

Changes in the tools and materials of education are paralleled by changes in orientation and methods. Stanford, like other universities, is emphasizing the extension of quality education to minority and disadvantaged groups. There is a growing interest in the social and ecological effects of technology, and in the improvement of our society in the face of social turmoil. Colleges and universities are being called on to divert resources from traditional channels and disciplines to new, problem-oriented programs that are often interdisciplinary. Students are encouraged to make broad and independent use of the intellectual resources of the university. Faculty members require up-to-date information to support research into the social and economic problems of the nation. It is generally acknowledged that today's most critical problem in the academic world is the limited resources available for fulfilling these new responsibilities. Networks and shared facilities, systems based on new technology, plus efforts to promote standardization, appear to be the only practical remedies in the current environment.

The role of the library is to support these constructive changes in American higher education with efficient use of its operating dollars and with improved services. The information requirements of the university are changing rapidly; the services of the library must change to meet these requirements. The creation of an on-line library automation prototype (BALLOTS I) was a major milestone in this effort.

1.4 OBJECTIVES

The overall objective of Project BALLOTS is to improve the library's contribution to university education through the application of computer technology to library processing. The large library is a production processing system that acquires library materials and makes them available to users. Library materials and inquiries about them represent a large daily volume in both the technical processing and circulation areas. Costs and quality of service are a function of the speed and accuracy with which the procedures and files of the library support a highly trained staff. BALLOTS II focuses on improving this support with an on-line production processing system. It is expected that the results will include improved quality of service; more powerful search facilities; system flexibility; far fewer files to maintain; automatic production of management reports and statistics; economic utilization of Library of Congress cataloging; better control over materials being processed; shared files and original cataloging among the institutions using the same system; and some reduction in the inadvertent duplication of purchases among network institutions.

The key to a library's store of knowledge is its file of bibliographic records. The multitude of paper and card files used to order and prepare books for use, control their circulation, and provide bibliographic information (through the card catalogs) to users are expensive to maintain and increasingly cumbersome to use. The machine-readable files created by BALLOTS II will be more responsive to the users and to the library staff in three ways. (1) They will be more up-to-date because of the speed with which changes to individual records and to an entire group of records can be made. (2) Each record will have a greater number of access points than exists for the present individual record stored in a manual file. (3) Several users will be able to use the same record simultaneously, and BALLOTS II files will be accessible to remote terminals. It will be increasingly unnecessary for the user to come to a file, since the file will be available wherever a terminal is located.

Accurate knowledge of the status of library material before it is available on the shelves is necessary for both librarians and library patrons. Accurate knowledge of what books are on order assists in avoiding unnecessary duplication. Accurate knowledge of where a book is in technical processing enables the library to advise a patron when a book will be available and to provide rush service when needed. With a manual system such information is difficult and time-consuming to obtain. BALLOTS II files will provide the librarian and the patron with immediate status information.

The growth of interdisciplinary research and teaching is placing a heavy demand on library resources and frequently results in undesireable, unnecessary duplication of library materials--a situation that is difficult to tolerate when book and periodical prices are rising at annual increments ranging from 10 to 20 percent. Centralized machine control of materials purchased for many academic departments and branch libraries is expected to aid in controlling these costs. Of particular assistance will be the ability to search files rapidly from remote locations and to communicate the results more rapidly to the patrons there. The value of such automated record keeping and rapid communication is incalculable to a network of cooperating libraries, because it offers the possibility of more rational sharing of resources--particularly book budgets.

The proposal of March 1970, which requested funds to begin BALLOTS II development, stated the following objective. Although the necessary funding to complete Project BALLOTS as originally scheduled was not continued after the end of this reporting period, and thus only "part 1" of Phase II was completed, brief conclusions have been inserted below after each component of the main objective.

"The main objective of Phase II of Project BALLOTS is to apply the learning from the completed basic research and prototype operations in Phase I to the creation of a fully operational, computerized system for the bibliographic management of the large library system. The components of this basic objective in Phase II are as follows:

> "Reliability, file security, and rapid recovery from downtime must characterize the operational system."
>
> <This component is being developed and will be tested in operations. In order to improve the reliability of the file and to allow for file recovery, the following design has been adopted. Whenever a new record is added to the file or an existing record is modified in the file, a copy of the new or modified record is placed in a separate data set called a "deferred update queue." Once a record has been placed in the deferred update queue, any user attempting to reach the old record in the main file will automatically be routed to the modified record in the deferred update queue. Thus the user sees the

latest version of the record and is unaware that the main file has not yet been physically updated. To meet file security and file recovery requirements, two copies of the deferred update queue are kept on-line on different physical disk drives. At night when the on-line system is no longer operating, the deferred update queue will be dumped to a tape and the on-line main file will be updated with the new and modified records. The on-line file indexes will also be modified at this time.>

"The operating costs of a production-engineered system must be less than or reasonably competitive with the costs if a manual system had been retained."

<Preliminary cost calculations show that the on-line system will be more expensive than the present manual system for a period of approximately five to ten years. After this period of time, the automated system is expected to be less expensive than the manual system. One of the reasons for the time lag is the fact that displaceable costs do not become cash savings until the library has been able to reduce staff through "attrition." If the benefits derived from the displaceable costs are considered as savings, then the break-even point would be reached much sooner. Another unknown factor at this time is the income from the network of libraries that will use BALLOTS. Since the system is not operational as this report is being written, actual measurements and costs are not available; high-cost calculations were therefore used to determine the operating costs of the system. It will be possible to calculate costs much more accurately after several months of production operation.>

"Multiple data bases and multiple users must be served. As expressed in earlier proposals and reports, centralized, computer-maintained files should be remotely accessible."

<This component is assured. Both the BALLOTS I and BALLOTS II designs have permitted access to a number of data bases and access to the same data base concurrently by a number of users. The BALLOTS II MARC file can be used by several typewriter and video terminals simultaneously with no noticeable degradation in service.>

"The system must be generalizable for external transfer to other institutions or regional utilities, such as library regional processing centers operated either by the private or by the public sector."

<This concept is to be tested by the California Library Automation Network--see section 3.7. One of the requirements that must be met by each library in the network is that the library must review, revise as necessary, and accept the specifications for each module. As this report goes to press, several of the network libraries have reviewed the BALLOTS-MARC requirements documentation; these have requested relatively insignificant changes to accommodate their use of BALLOTS-MARC. This experience has been most encouraging, and if the remaining modules follow the pattern of BALLOTS-MARC, this objective will have been met.>

"User requirements rather than machine convenience should continue to dominate the design."

<This has been true and will be tested in operations. A major design task has been designing the user/video-terminal interface. The requirements for CRT terminals were determined as a function of the user interface; and only after these requirements had been drawn up was a CRT terminal sought that would satisfy them. Sections 2.4, 3.2, and 3.3 of this report describe this work. A programmable terminal was chosen because none of the available hardwired CRT terminals would satisfy all of the criteria established by the BALLOTS project. The features that have not been found in existing CRT hardware will be added as software to make the system as convenient and flexible as possible for its users.>

"Service levels for processing transactions must be improved--not just maintained at the present level--in the face of increasing work loads."

<This component is assured. Once the system is in operation, it will be possible for the processing load to grow substantially without requiring additional manpower. We expect that the terminal operators will be able to handle a significantly heavier work load under BALLOTS than they are able to under the present manual system. In addition to this, the BALLOTS system can be expanded to add new modules that make possible new types of services, unavailable under the present manual system. It is expected that much of this expansion of services can also be done without additional manpower.>

1.5 SHARED SOFTWARE

Shared software is an important concept in BALLOTS and SPIRES development. BALLOTS is a data management system for libraries; SPIRES is an information retrieval system geared to the needs of researchers in many disciplines and involving many data bases. Both applications require substantially the same basic software for file services (file building, update, creation of indexes), retrieval (parsing, decoding of command languages), and output (printed reports and CRT display of search results). In the BAL1OTS prototype system, the following was achieved through common software and described in the BALLOTS I Final Report

1. On-line, interactive searching permitted users to conduct file searches in which they could easily modify, expand, and contract search requests, and save the early, intermediate, and final results. An operator could complete a typical complex search within one or two minutes, a simple search in a few seconds.

2. Several data bases were serviced by this software. This feature was implemented on five different data bases: the University Libraries' In Process File, ERIC, preprints in high-energy physics, and two individually maintained files, one on African history and one on geology.

3. Several users could search simultaneously the same or different files without interference to one another.

4. The sample outputs from library technical processing were demonstrated for search results, purchase orders, and other printed forms.

5. Computer or programming knowledge was not a prerequisite for users; productive results could be obtained after a few hours of training and experience.

6. The user was able to communicate his satisfaction or dissatisfaction with the on-line system easily and directly during a terminal session. Thus the user became part designer and his feedback influential in arriving at the production system.

To the above list BALLOTS II adds file integrity and file recovery software (see the description of this in section 1.4).

1.6 DEVELOPMENT STATUS

The prototype system, BALLOTS I, was implemented, operated, and evaluated in 1969. This provided the foundation and experience to design and implement a production system, BALLOTS II, an effort that is still in progress. Details of the organization, staff, facilities, and results of the prototype system development are given in chapter 2 of the BALLOTS I Final Report.

In the period covered by this report, July 1970 through June 1971, user requirements were defined for major system processes, such as ordering, receiving, and cataloging (sections 2.1 and 2.2). Video terminals were evaluated and the Sanders PDS (804) terminal was selected (sections 2.4.8 and 3.2). Video terminal screen formats were designed (sections 2.4--passim--and 3.3). Host computers for BALLOTS II development and implementation were studied and one was chosen (section 3.1). The BALLOTS on-line command language was developed (section 3.4), a notation system for describing data elements in various formats devised (section 3.5), and the programming language selected (section 3.6). Programming basic software to support file services, on-line interactive searching, printed outputs, and recovery procedures continues.

When Office of Education support was terminated in the first half of 1971, it became necessary to alter the system development approach. With the single omission of serials, the original approach to BALLOTS II was identical to that for BALLOTS I: large-scale, system-wide applications to sequentially related functional areas--acquisition, cataloging, and circulation.* With the goal of achieving near-term operations within the more limited resources available, it was decided to adopt a modular approach. This had several advantages: automation could be introduced more gradually; almost all parts of the library--including potential network members--could begin participating in BALLOTS at once; and modules providing the most benefit to the greatest number of users could be introduced first. The modular approach is further detailed in section 2.3.

----

*Although serials were to be covered in the BALLOTS I design, this was decided against for BALLOTS II owing to library staff priorities and the nature of the justification for automating on-line. The following paragraphs explain the distinction made between serials control and the other areas of technical processing that led to excluding serials from BALLOTS II.

Much of the effort in recording serials lies in the manual handling of the several hundred thousand individual pieces

The BALLOTS-MARC module will be the first to be implemented (see section 2.3). BALLOTS-MARC will enable users to search the MARC file interactively from a variety of access points (personal name, corporate name, title words, and Library of Congress card number) singly or in combination. Implementation is planned for spring 1972; CRT terminals have been ordered, delivery dates specified, and a PDP-11 interface computer installed to service the CRT terminals.

Major tasks confronting the designers of bibliographic systems using time-sharing include: (1) identification of system requirements, (2) access to a machine with adequate resources to insure satisfactory service, (3) reasonable pricing to provide an affordable service, (4) rapid recovery from machine and system downtime, (5) complete file integrity, i.e., no loss of records, (6) selection of a video terminal suitable for the input and display of bibliographic records, (7) design of file services, on-line interactive software, and applications programs, (8) documentation, (9) user training, and (10) resource management. The methods for attacking these problems have been documented in the Final Report on BALLOTS I.

----

received each year--a task in which the computer offers no assistance. Additionally, the efficiency and economy of Stanford's manual serials control system has presented a consistent and strong obstacle to a move towards computerization. The department in the Main Library controls some 26,000 current titles, plus retrospective records; it functions with a small but dedicated nonprofessional staff, and its work is always up to date; it has suffered temporary backlogs only as a result of factors beyond its control, i.e., dock strikes or mail strikes. Procedures in the manual system are consistent, well designed, and well taught.

There are two areas in which some assistance may be derived from the computer, but neither requires an on-line response: one is file security--the current manual file is backed up only by a microfilm that records file status as of the date of filming. The second area is claiming, which is a housekeeping task of substantial nuisance value, but not unbearable, and can be served effectively through a batch system. The University is currently considering preparation of an expanded (and ultimately complete) printed union list of serials with generalized holdings. Coverage is presently confined to currently received science and technology titles. The completion of this task is believed to offer a more economical solution to effective public service as well as some assistance to security problems with serial records than would development of an on-line control system.

BALLOTS' experience has shown that the use of an on-line facility for managing the basic bibliographic operations of a large library and of a cluster of nearby libraries is feasible and sound. The technical problems are substantial but solvable, using a central, shared facility supporting both library and other users. Economical use of the available computing resources (including manpower) is the principal challenge to the designer.

We recommend that the BALLOTS project be carried to completion and that BALLOTS form the system for a regional network in Northern California; such a network has been designated CLAN--California Library Automation Network (see section 3.7).

1.7 RELATED RESEARCH

The expense and complexity of library system development have been recognized for several years and have been documented <2,3>. At one time it seemed natural to suggest that development work might be shared and that certain elements of system design be commonly developed by several institutions. Thus far such hopes have not been realized. Despite the attractiveness of the idea of collaborative development, it continues to elude researchers. Under a grant from the National Science Foundation, three university libraries--Chicago, Columbia, and Stanford--began a project in 1968 to test the feasibility of a collaborative design effort. Although the project did not achieve its goal of creating a unified system design for a target process (acquisitions), it did provide an understanding of the staggering proportions of large-scale system development in both the human and technical areas. The material in Appendix A, extracted from the Final Report on Collaborative Library Systems Development (CLSD), summarizes the conclusions reached by participants in the joint effort.

CHAPTER 2

BALLOTS II BIBLIOGRAPHIC SERVICES DESIGN

2.1 BALLOTS DESIGN REQUIREMENTS

BALLOTS is predicated on the application of video terminals for the following reasons: (1) patrons or librarians should be able to conduct searches rapidly and silently, so that others will not be disturbed, even in a public reading area; (2) facilities for self-instruction and for locating and correcting errors should be incorporated into library procedures, to minimize mistakes and increase throughput. With these goals in mind, the general features of the BALLOTS bibliographic service design are as follows:

1. Access throughout the normal working day to automated acquisition, cataloging, and circulation services.

2. Implementation as a series of modules (sets of interrelated services), with each module achieving stable production status before another module is implemented.

3. User interaction with the system via video display units in network libraries, in the Stanford library, and throughout the Stanford community. (The library files will also be accessible for searching via the more than 130 typewriter terminals on and off campus.)

4. In order to ensure reliable network operations, implementation of all files and services first at Stanford, to be released to network members only after thorough testing.

5. Four on-line files: a six-month to one-year file of the most recent MARC data, an In Process File (IPF) of all titles on order or in technical processing, a Catalog Data File (CDF) of all titles cataloged, and a Circulation Inventory File (INV) of all titles in Stanford's Meyer Undergraduate Library collection.

6. Multiple indexes for each file, such as author, title, and LC card number, and an easy-to-use command language with Boolean search capabilities.

7. Printed outputs: purchase orders, technical processing control forms, catalog card sets, book spine labels, and management statistics. These will all be available from the implementation of the first module. Users will be able to create bibliographies from selected portions of files. It will be

possible to produce lists of new acquisitions by subject and to send them to special user groups or academic departments.

8. Reliability, fast recovery, and file protection as an inherent part of the software-hardware design.

2.2 LIBRARY SYSTEMS ANALYSIS

The bibliographic services design originally conceived in BALLOTS I concentrated on the functional aspects of library technical processing: a series of complete subsystems for acquisition, cataloging, circulation, and serials control. Each major subsystem was broken down into a series of processes. The acquisition subsystem consisted of ordering, receipt of purchase order materials, receipt of non-purchase order materials, invoice receipt, dealer report receipt, automatic claiming, cancelling, forced claiming and cancelling, and MARC conversion. The catalog subsystem consisted of distribution and MARC searching, Library of Congress (LC) cataloging, original cataloging, Meyer cataloging, added copies and volumes, and maintenance. The circulation subsystem consisted of lost book billing, charging, circulation search, discharging, delinquency, fine payment, fine search, hold/recall, initial check-in, missing, overdue, and patron search. The reserve processing subsystem consisted of reserve book processing, reserve ordering, reserve search, reserve book listing, and off reserve.

During the reporting year, the acquisition and catalog subsystems were fully documented. In addition to the five BALLOTS analysts, three members of the library staff worked on this documentation at the project site for three months. The documentation comprises a process book for each process in each subsystem plus a system book common to both the subsystems. These assembled volumes contain over 1,000 pages.

A process book begins with a flow chart and a narrative description of the process. Then follow a series of forms describing each of the inputs and outputs of the process, the video terminal screen formats and the files used in the process, and the manual and manual-automated procedures performed in the process. Appendix B contains the narratives and flow charts from two of the acquisition and two of the cataloging process books produced. Statistics were gathered from library staff, and estimates of the volume of printed outputs, searches, video terminal usage, file updates, etc., within each process were made. As the process books were completed, they were reviewed by supervisory librarians in the appropriate departments, and changes requested were incorporated into the documentation.

The system book contains data that is common to all of the processes. For instance, an input screen format may be used in both ordering and cataloging. The format and the editing rules that apply to it are defined in the system book; the descriptive, procedural, and statistical information about a process using this input format are documented in a process book. The system book consists of library requirements translated into explicit specifications from which programmers develop program specifications. It contains, for each screen format or printed output, a sample format, the data elements contained on it, and a set of processing rules that explicitly define line and column positions for display and printing, editing rules for input screens, etc. The system book also contains, for each system file, the data elements contained in the file and processing rules for input, update, and searching. Processing rules for the editing of inputs and printed outputs are written in BALLOTS data element notation (BDEN) (see section 3.5). Library staff also worked with the BALLOTS analysts on the requirements contained in the system book. The system book was reviewed by both the library staff and the BALLOTS programmers for accuracy and feasibility. Sample pages from the system book are also included in Appendix B.

A major result of analysis efforts during the year was a sophisticated bibliographic file organization, capable of managing the complex records of materials in process, yet easy for the library to use. Three major factors influenced this work: (1) the need in technical processing (particularly in automatic claiming) for being able to keep track of individual physical items; (2) the availability of the analyzer and the parser developed in Project SPIRES <4>; and (3) the development by SPIRES staff of a file definition language.

The In Process File (IPF) is a prime example of this file organization. The four structural levels of the IPF are shown in Figure 1. The bibliographic structure contains the full bibliographic data for the title. The library structure contains library-specific data (such as a varying call number) for any library that is ordering or holding that title. The acquisition structure contains data about a specific order by the library for that title: vendor, accounting data, requester information, etc. The item structure contains control information for each physical item ordered, such as date of order, receipt, claim or cancellation, and invoice data. When the material is cataloged, the acquisition structure is deleted, and the item structure is replaced by a holdings structure that gives the shelving location and copy number of that physical item.

Within any given bibliographic structure in the IPF there is one library structure for each library ordering that title. Within each library structure there is one acquisition structure for each order placed by that library. Within each acquisition structure there is one item structure for each physical item represented in the order.

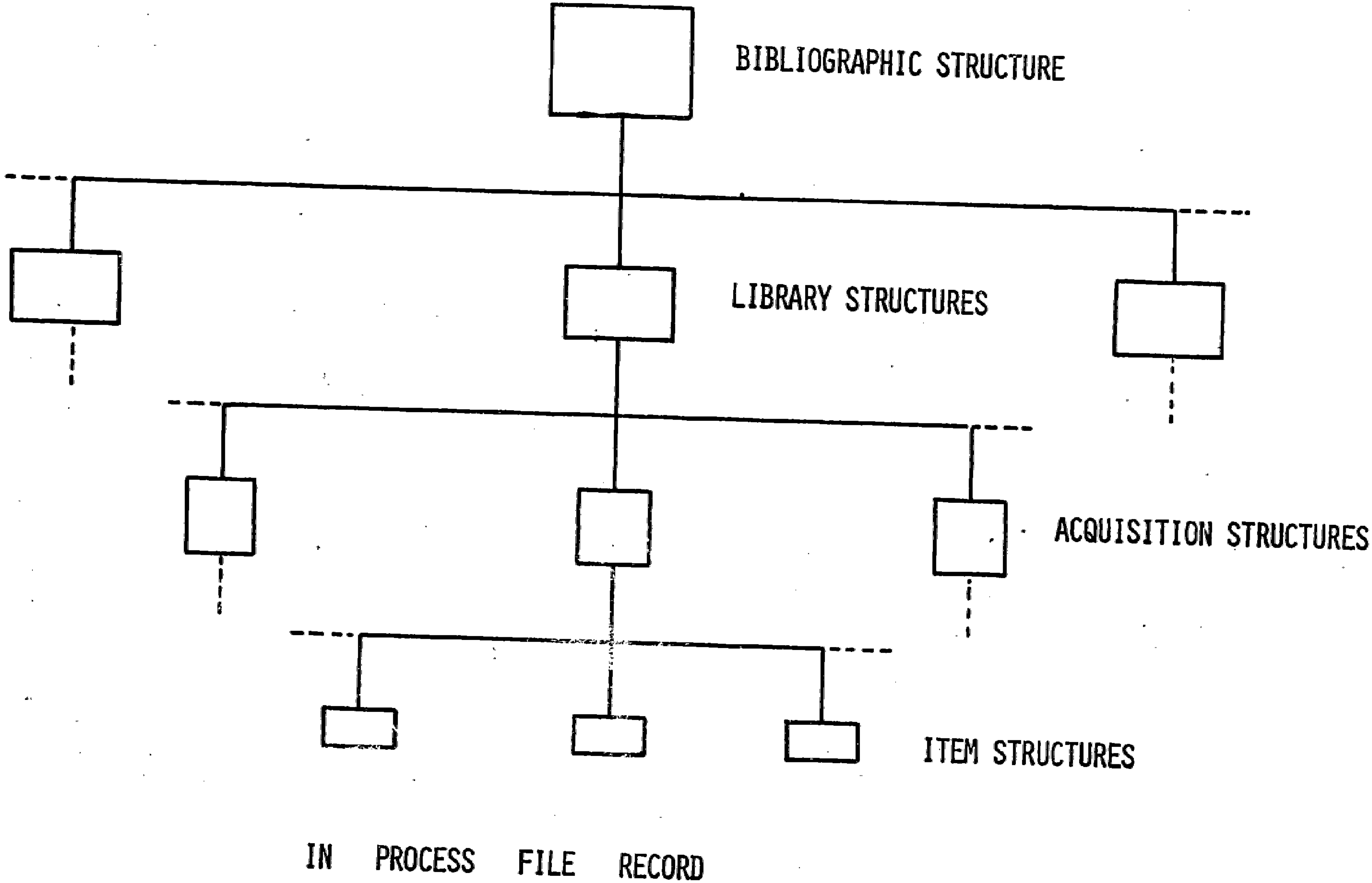


Figure 1. In Process File Structure Levels

A command language syntax was developed that would enable a terminal operator to create a number of item structures through a single input string. For example, if an acquisition operator were to input the string "3c (volume 1, part A; volume 2; volume 3, part A-C; volume 4-6)," the following 24 item structures would result:

```
volume 1, part A   (copy 1)
volume 1, part A   (copy 2)
volume 1, part A   (copy 3)
volume 2           (copy 1)
volume 2           (copy 2)
volume 2           (copy 3)
volume 3, part A   (copy 1)
volume 3, part A   (copy 2)
volume 3, part A   (copy 3)
volume 3, part B   (copy 1)
volume 3, part B   (copy 2)
volume 3, part B   (copy 3)
volume 3, part C   (copy 1)
volume 3, part C   (copy 2)
volume 3, part C   (copy 3)
volume 4           (copy 1)
volume 4           (copy 2)
volume 4           (copy 3)
volume 5           (copy 1)
volume 5           (copy 2)
volume 5           (copy 3)
volume 6           (copy 1)
volume 6           (copy 2)
volume 6           (copy 3)
```

Rules for such compressed input strings were written in modified BNF (Backus-Naur Form) notation to be passed through the SPIRES "action" analyzer. The syntax definition for these rules was generalized enough to accommodate variations in the format and content of the input string. For example, the following lines are equivalent:

```
3 (v 1)
3 c. (vol. 1)
3 c (volume 1)
3c. (v. 1)
```

This generalization was done to give the operator as much flexibility as possible in composing input strings.

The presence of item structures enables the computer to deal with individual items in technical processing. For instance, if a partial shipment of a library order is received, the operator uses a matrix-like display screen format to check off the items received. The system then generates a catalog data slip for the material received. This slip gives status information on the entire order. During a weekly automatic claiming cycle, the system will also generate a claim notice for the items ordered but not received.

2.3 CONVERSION TO THE MODULAR APPROACH

A reduction in available external funding necessitated an alternate development scheme for achieving the project's goals--modular implementation. In this approach, functional modules that cut across major subsystems are developed first and implemented sequentially. For instance, acquisition and cataloging both need to use MARC data--therefore, by implementing a MARC utilization module, it is possible to spread system benefits throughout the library almost at once, without delaying the next subsystem--cataloging--until completion of the first, and without the relatively greater complexities of overlapping two major subsystems. (The MARC module also offers the advantage of more immediate benefit to potential members of the CLAN network.) Additionally, because a module cuts across many functional areas, a wide base of users is exposed to automation very quickly, and experience gained in the operation of the first modules can be applied to the later ones.

The documentation required for the modules can be derived almost wholly from the process and system books that had been prepared for the original bibliographic services design. The module documentation is organized in much the same way as the full system documentation, using the same components.

The eight general features enumerated in section 2.1 have been embodied in a series of modules that may be thought of as sets of services providing progressively expanding service for a widening variety of library material. There are 11 modules; they are described below in the order in which they will be implemented. The detailed requirements for the first module have been completed, documented, approved by the Stanford Library, and are in the process of being approved by each CLAN library.

1. BALLOTS-MARC. The library material processed by the MARC module is English-language monograph material appearing on weekly MARC tapes. (The restriction to English-language material is a consequence of the current scope of MARC; all roman alphabet languages are supported by this module.) The file in this module is an on-line MARC file of the most recent 6 to 12 months of MARC records. The file is essentially read-only except for the addition of usage and date codes for records processed by users. The actual size of the on-line file will depend on the requirements of the network libraries balanced against file

storage costs. Purchase orders, process forms for technical processing files, catalog card sets, and spine labels will be produced on request for any titles in the MARC file. Automatic weekly searches to match user requests with new additions to the file will be available through a standing search feature. In this first module no permanent on-line records will be maintained during technical processing other than the full MARC record and its usage status and date codes, although a tape copy of the records for each book cataloged will be retained for later use. Such on-line record keeping will appear in modules 2, 5, and 6. This module will process approximately 35 percent of Stanford's acquisitions and 26 percent of its cataloging. The percentage of support to CLAN processing is slightly larger than the Stanford figures in this and later modules. The programming of the BALLOTS-MARC module is being accomplished with Stanford's own financial resources.

2. MARC-IPF (In Process File). This module adds an IPF and additional printed outputs such as claim and cancellation notices, when requested by library staff. Only MARC material is handled; when a record is found in MARC it is transferred to the IPF and is retained there as an updateable record throughout technical processing. Since the record will not be purged from the IPF until modules 5 and 6 have been implemented, the file will represent all titles ordered and cataloged by the library using the automated system. A record in MARC-IPF can be used again if additional copies of a book are ordered.

3. Purchase Order/Original Cataloging. No new file is added with this module, but the use of the IPF file is expanded considerably. Also, Title II Slip and National Program for Acquisition and Cataloging (NPAC) notices can be produced. The scope of material for which a record is created is expanded considerably. It adds all non-MARC roman alphabet material that requires a purchase order in ordering, and any material that requires original cataloging. Thus, if a record is not found in MARC, a new IPF record is created on the terminal. This module will process an additional 52 percent of acquisitions and 42 percent of cataloging. Thus services at this point will cover 87 percent of acquisitions and 68 percent of cataloging.

4. Non-Purchase Order Material. The scope of material added to the IPF is expanded to include non-MARC non-purchase-order material receipt--gift, exchange, approval, and blanket orders. In addition, an invoice claiming feature is included to inform the Acquisition Department of material for which no invoice has been received within thirty days. This module will process an additional 7 percent of acquisitions and 6 percent of cataloging. Modules 1 through 4 will process a total of 94 percent of acquisitions and 74 percent of cataloging.

5. Catalog Data File. This module involves building the on-line Catalog Data File. Since the implementation of module 1, BALLOTS will have saved bibliographic information, and this data will be used to create the CDF. From this point on, all catalog records will enter the CDF after the record for a given title is no longer required in the IPF. As the CDF grows, it will become an increasingly valuable reference tool for acquisition, cataloging, and patrons' use.

6. Inventory File. Machine-readable bibliographic and holdings records already exist for all 60,000 titles now in the Meyer Undergraduate Library. In this module, these records will be converted to BALLOTS format and used to build an on-line Meyer Inventory File (INV). At this point, Meyer cataloging processing will work directly with the on-line file. This file will be used later on for reference and for the patrons' access to the complete holdings of the undergraduate library. Other libraries with the entire collection in machine-readable form can be handled in a similar manner.

7. Book Catalog. This module can be used to create any book catalog done in the Stanford format. At Stanford it will allow the Meyer Book Catalog to be produced directly from the INV without going through the punched card process presently used.

8. Automatic Claiming and Cancelling. This module adds programs to review IPF records automatically, to determine if ordered material is overdue. Material may be claimed several times and finally cancelled if the dealer does not respond. The Acquisition Department may override a scheduled claim or a cancellation.

9. Circulation. This module is designed to handle the complexities of the research library circulation system. Using data from the Inventory File, a Meyer Library self-service circulation system will be implemented first, including charging, discharging, initial check-in, circulation searching, recall, holds, overdue processing, fine handling, and fine payments.

10. Standing Order and Out-of-Print Desiderata. The capability of establishing standing orders (SO) and receiving the non-serial materials arriving with SO's will be added with this module. In addition, out-of-print items (OP) will be added to the IPF, and search and quote letters produced for OP dealers. If an OP item can be procured, it can be ordered using the record already in the IPF.

11. Reserve. This module adds reserve book ordering and processing for users. It will be added to the services offered to Meyer staff through the use of the INV and IPF.

Although modules 6, 7, 9, and 11 will be implemented first at Stanford's Meyer Undergraduate Library, each module will be generalizable and its application not limited to Meyer Library. Modules are not dependent on size of library--they can be employed in small college libraries, research libraries, and university branch libraries.

No module will be released for use at a member library until it has been in heavy production use at Stanford for a four- to eight-month period. This is done in order to insure reliable operational use of each module prior to installation in a network library.

Module development and implementation are discussed further in chapter 4, as these have been the major concerns of Project BALLOTS in the time following the reporting period.

2.4 DEVELOPMENT OF BIBLIOGRAPHIC OPERATIONS ON A VIDEO TERMINAL

This section summarizes the project's design philosophy for on-line interaction between the operator and an automated bibliographic system via a video terminal. It includes a discussion of the development of CRT terminal screen formats, the definition of protocols and a command language, and the final choice of a CRT terminal.

2.4.1 Background

Early in BALLOTS I, work was begun to identify CRT terminals suitable for the user-system interface. Unfortunately, equipment of sufficient reliability and economy that also met the functional requirements of a bibliographic application did not appear on the market as a production unit during the lifetime of BALLOTS I. Various attempts were made to use off-the-shelf CRT terminals, but none could meet the library's specifications. An account of this early work appears in the BALLOTS I Final Report. Although these investigations did not yield a suitable CRT terminal, they did provide important background that later activity built on.

By August 1970, system programming had progressed to the point where it was mandatory to think in specific detail about the way bibliographic data would pass back and forth between the operator and the system. Since no CRT terminal had yet been selected, a hypothetical terminal was projected as the basis for preliminary design. This procedure subsequently proved to be extremely valuable. The hypothetical terminal had the basic features that could be found on almost any machine. It was rather like a basic IBM 2260: very few functions, only uppercase characters, no protected format capabilities, a relatively small number of characters displayable at one time, and limited editing capabilities.

2.4.2 Transferability of Design Experiences with Typewriter Terminals

The design experiences derived from work with the typewriter terminals during BALLOTS I were not directly transferable to a CRT terminal environment. Communicating via a CRT terminal is entirely different from communicating through a typewriter terminal. A CRT terminal can transmit an entire block or screen of data at one time, a screen on which nearly 2,000 characters can be edited and corrected and forwarded as a whole. This represents a very different mode of operator-terminal interaction than does the typewriter terminal, which operates in a character-by-character or line-by-line transmission mode. The CRT affords no hard copy, and one cannot back up and observe what was input five minutes ago. The design staff had many years of experience working with typewriter terminals. Now they had to acclimate themselves to a new environment.

2.4.3 Formatted Screens

Given the model of the hypothetical terminal, the design staff began to develop CRT terminal screen formats. The object of formatting the display of information was to ensure that the operator would find the data elements consistently presented in the same form and position on the screen. Such consistency makes it easier for the operator to recognize various data elements, to develop efficient keying procedures, and to identify errors. When a particular element is always found in the same place on a screen, the operator need only check that one location, rather than reading the entire screen in order to determine if that element is contained in a given record. Similarly, if the input field for a certain element is always on the third line, the operator can key that data without having to look at the screen to position the cursor.

Since all 120 of the data elements possible in a BALLOTS record cannot be displayed at one time, it was necessary to divide the data elements into smaller logical groups. Each group represents the raw material for one screen format. Grouping was done on the basis of the logical relation of data elements, the work flow of the operator, and the length and frequency distribution characteristics of data element values.

Certain data elements have meaning only in conjunction with other data elements. The format for display of bibliographic data, which was modeled on the form of a Library of Congress entry, is an example of a format where data elements are recognized and interpreted on the basis of their order and position.

Some data elements must be grouped together because they are all required by the operator and/or the program to carry out a specific operation. If an operator is to make a decision to cancel an order, he must first know when the book was ordered, who the vendor is, what type of material it is, and whether or not the library has received any communication from the vendor concerning this order. Since the same data element may be necessary for several operations, some data elements appear in more than one format. For instance, the purchase order number must appear on a screen for cancellation of an order as well as on the screen for receipt of material on an order.

The design team decided that for the data input format, the system should prompt the operator with field tags. Experience gained in BALLOTS I indicated that it was preferable to have the system supply tags and have the operator fill in the blank fields, rather than to have the operator supply the tags along with the input data. This method of prompts places the burden of remembering tags on the system, and also saves keying. The tagging scheme is based on the data element mnemonics developed in BALLOTS I. (These mnemonics are described in the BALLOTS I Final Report, section 2.5.1.)

Data element mnemonics and input fields were positioned on data entry screens by considering operator work flow and the frequency of use of each element. In general, fields were positioned in the order in which the operator would generally use them. However, seldom-used elements were pushed to the bottom and to the right of the screen so that the operator would not have to key past something he would use only infrequently.

The use of prompted mnemonic tags also pointed to the desirability of field protection capabilities. A protected field is an area into which the operator cannot move the cursor and therefore cannot write, change, or erase data. The purpose of a protected field is to preserve system-supplied information (such as the format of tags and input fields), to reduce the possibility of entering data into the wrong field, and to facilitate cursor positioning for input keying. The cursor can jump automatically from the end of one input field to the beginning of the next. Tab functions can also be used to move forward or backward from field to field.

2.4.4 The Overflow Problem

The most perplexing problem in format design was caused by the wide variation in the length of certain bibliographic data elements. For example, an analysis of 500 personal name main entries taken from a MARC tape showed a minimum length of 6 characters and a maximum length of 53. The mean length was 21.74, and 90 percent of the lengths were less than 31 characters. A similar analysis of title statements showed a miminum of 4, a maximum of 191, and a mean length of 40.53 characters. Over 90 percent were less than 75 characters long.

Given the limited amount of space available on a CRT screen, it is not at all efficient to define input fields the size of the maximum length of widely varying data elements. Most of the time a format would be filled with unneeded blanks. Fewer data elements could be grouped into each format, and therefore the operator would have to use many more formats to carry out a given function.

In working with the hypothetical terminal, it was concluded that extremely variable data element values should be handled by a combination of fixed-length fields and an overflow area. The overflow area might possibly be at the bottom of the format, or it might be a separate format. The fixed-length fields were designed to accomodate 80 to 90 percent of the occurrences of such an element. This was a cumbersome prospect for both the programmer and the operator.

It became apparent that the only satisfactory solution to the overflow problem was to provide expandable fields for these troublesome data elements. It was essential that this expansion take place without sacrificing format protection. And it was desirable that the necessity for expansion be detected and take place automatically, without any special intervention on the operator's part.

2.4.5 The Role of Data Element Statistical Analysis in CRT Screen Design

In order to group data elements within protected formats, it was necessary to have specific information about the behavior of those elements, both in terms of size and occurrence. The range of size of each element affected the size of the input field, and the relative occurrence of elements affected their ordering within a format.

Both the Library of Congress and Columbia University had performed extensive analyses of MARC data to determine the frequency of occurrence of certain data elements and graphic characters, as well as the average lengths of data elements <5, 6>. However, to design a screen format for display of bibliographic records BALLOTS required more information than just average length. Therefore, a statistical program was written to learn the distribution of length of bibliographic data elements. The purpose of this study was to determine how best to deal with the limitations of terminal hardware requiring fixed-length data fields, as well as to aid in file design. A program was written to profile the distribution and length of each data element in

MARC records and in the Library's machine-readable In Process File (IPF), prepared during the operation of BALLOTS I. This program reads MARC and IPF records that have been transformed into the BALLOTS file-building format. The program accepts an input parameter specifying the mnemonic of the data element to be analyzed. It analyzes and tabulates the length of the data element in the record. The following summary information is printed for the data element in the batch of records examined:

- Number of records examined
- Number of occurrences of data element
- Number of entries with no occurrence of data element
- Number of entries with multiple occurrences of data element

For each data element, the following information is tabulated:

- Length
- Incidence
- Incidence sum
- Incidence percentile

For each data element, the following statistics are calculated:

- Minimum length
- Maximum length
- Mean length
- Variance
- Standard deviation
- Skew
- Kurtosis

As an example, in the MARC tapes dated August 13, 1970 and October 1, 1970, the following information was obtained for the data element MEPN (Main Entry, Personal Name, equivalent to MARC tag 100):

Date	8/13/70	10/1/70
Length for incidence percentile of 90 or greater	31	33
Number of records examined	500	500

In checking the IPF for the data element VSP (Special Instructions to the Vendor), 106 occurrences were found among 700 records. 90 percent of the instances were accommodated in four characters; the next length, 12, occurred only twice; length 36 occurred twice; and six other lengths ranging from 13 to 59 characters occurred only once each. Appendix C contains examples of the output of these programs. The output from this statistical program was influential in arriving at the BALLOTS file and screen design.

2.4.6 Practical Experience

While still designing with the hypothetical terminal, the design staff examined the Sanders 720 CRT terminal that was being used at Stanford's Administrative Data Processing Center. This CRT terminal did not meet project requirements on a number of grounds: most notably, it did not have an upper/lower-case character set. However, it did provide format protection and give the designers concrete experience. Various screen formats were built and displayed to the development staff and to librarians. Feedback revealed problems and suggestions for changes.

During this period, the design staff took advantage of other opportunities to work with various CRT units. Formats were set up and altered to take advantage of a particular feature or to get around a particular problem inherent in each different terminal. Library staff members worked at the terminals to determine what kinds of errors might be made if data were organized in a certain way, or what mistakes could be avoided by a different organization of the data.

This experimentation facilitated the move from a general to a more specific set of terminal requirements. Some of the ideas originally developed for the hypothetical terminal were found completely untenable when a real terminal was available. For example, the first solution to the overflow problem (leaving blank lines at the foot of the screen) turned out to be extremely difficult to use. Some of the visual organization of data that had looked fine on paper was unsatisfactory on a screen which lacked the niceties of grid lines to illustrate columns and rows. Operators found it very difficult to orient their eyes to such a screen. Also, when a cursor became available, the problems inherent in moving it about from one position to another required the rearrangement of some fields so that they were in more efficient positions or required less cursor movement.

2.4.7 Development of Protocols and Commands

To carry out a given function, such as ordering a book, the operator must execute a prescribed set of optional and required actions involving a specific set of screen formats. For example, the operator must first determine whether or not the book is already on order. His first activity is searching. That might be followed by the use of a second screen format for displaying the search results. As a third step he might wish to amend existing bibliographic data. A fourth step requires the addition of data elements needed for acquisition operations, e.g. vendor, purchase order number, requester, etc. Finally the operator must signal the system that the record is complete and ready for final machine editing before it is entered into the data base. The order and interrelationship of these actions should be designed to provide the operator with as much flexibility as possible. This order and relationship must also be clearly defined so as to ensure completion of the function being carried out.

The map of legal paths through a function is called a protocol. The term "protocol" is used here in a larger sense than usual. It is common to talk about communication protocols between the terminal and the computer that cover polling strategies, interruptions, and data communication commands such as "request for transmit," "negative acknowledge," and the like. However, the design staff felt it important to think in terms of a higher level of protocol that included the operator's actions. From the project's viewpoint, the most important part of the entire communication chain is what the operator wants to do.

The objectives in designing a protocol are to optimize the normal sequence of actions, to enable the operator to deal with any exceptional situations that might arise, and to disable all actions that are extraneous or detrimental to a given function. All possible paths through a function are mapped so that one always knows the status of each piece of data at any given moment; so that it can be made clear to the operator which element of the system is in control at any point in time. For example, if the operator wishes to correct an error, he needs to know the exact status of the data in question, whether corrections can be done directly by using the editing features of the terminal, or whether the information has already become part of the permanent data base and therefore requires a more elaborate correction procedure. The system is designed to make it as evident as possible what can, cannot, should, and should not be done at a particular point in a protocol. The operator should not have to waste time trying to decode the status of his data.

While the protocols were being defined, a command language was formulated to drive the system through the protocols. As the operator issues various commands, he chooses one of the many possible paths through a protocol and satisfies the requirements of that protocol. The command language provides the direct means by which the operator instructs the system as to his wishes and intentions. The protocols by themselves are passive models; it is the command language that initiates action.

The operator is prompted with the commands for the main line, the most common route through a protocol, as a default option. Each screen format contains a command field in which the system supplies the command verb that will produce the next step in the main line of that protocol. Thus the operator does not need to take any special actions to deal with the usual cases.

The command prompts are a function of the protocol, and are independent of the particular screen format in which they appear. The same format may appear for several different protocols. For example, the format for input of basic bibliographic information is needed to produce a purchase order and also to produce a set of catalog cards. In other cases a format that is required in the main line of one protocol may be an optional branch in another.

The operator uses the command language to tell the system which function he wishes to use, e.g. ordering, receiving, original cataloging, reserve processing, etc. The system then enters the operator in the protocol specifically designed for the function he has requested. This enables the operator to accomplish his particular purpose without having to request over and over again the facilities he will require on a routine basis. The protocol also serves to deny to the operator formats and command actions that are inappropriate to his function.

The command language is also the mechanism that allows the operator to instruct the system to take one of the options in a protocol. When the operator needs to depart from the main line, he simply overwrites the prompted command with some other command. For example: if an operator is using a series of formats to input both bibliographic and ordering data, and realizes at the end of the protocol that something was wrong with the bibliographic data he has already entered, he can, in effect, tell the system: "I have completed the ordering data and it is correct; but before adding this record to the file, I need to back up and correct bibiographic data input on a previous format in this protocol." To do this, the operator simply overwrites the prompted command that would enter the record into the file, and replaces it with a command instructing the system to redisplay the format containing the bibliographic data he input earlier in the protocol. Command options such as this not only ensure that the operator can handle all situations; they eliminate the need to re-key any data other than that which is in error.

The command language was developed with the aid of a BNF analyzer program that checked all command combinations for internal logical inconsistencies. The staff also crosschecked the BALLOTS command language with other command languages in use at Stanford for inconsistencies between languages. Thus, as a user moves from one system to another, he will not be confronted by a situation where the same command verb is used to mean different things in different systems, or where different commands are used to mean the same thing.

2.4.8 Terminal Hardware Evaluation and Selection

Armed with the above design experience, the BALLOTS staff was prepared to make an intelligent and thorough analysis of terminal equipment. Therefore, early in 1971 they began an intensive review of all available CRT terminals that seemed likely to meet project requirements.

Eight major factors governed the choice of the CRT terminal for BALLOTS. These, or analogous constraints, are likely to be present in any project. They were as follows:

1. The Host Computer

   The terminal had to be compatible with the hardware and software environment of the Campus Facility's IBM 360/67 and the front-end PDP-11 used to serve CRT terminals.

2. Generality

   The terminal had to be useful for other applications in the academic community served by the Stanford Computation Center.

3. Data Communications Mode

   The terminal had to support asynchronous, block mode communication at 9,600 baud. Asynchronous transmission was required to accommodate the Campus Facility's locally produced modems, which had been obtained to eliminate the rental cost of equivalent units from common carriers. The requirement for block transmission at 9,600 baud was based on efficient use of machine resources and existing communication lines.

4. Character Set

   Because the data base and all printed outputs were to be upper and lower-case, the terminal had to have such a character set. It is important to note here that a decision was made by the Catalog Department of the Stanford Libraries to do without diacritical marks and other special characters. It was felt that the additional cost of keying, displaying, and printing the special characters, as well as of their inclusion in the terminal equipment, did not balance against their added utility and benefit.

5. Full Editing Capabilities

The terminal had to have full cursor controls, including: up, down, left, right, home, and tabs. It also had to have character insert, delete, and overstrike, and line insert and delete.

6. Screen Capacity

The terminal had to be able to display at least 1,000 good quality characters at one time. A capacity of 2,000 characters was preferred.

7. Format Protection

The terminal had to have field protection capability, and some method of field expansion to handle overflow.

8. Practicality

The terminal had to be available at an affordable price, reliable, made by an experienced manufacturer, and supported by a good service organization.

Two developments in the technology of terminal manufacture provided choices in 1970-71 that were not available when Project BALLOTS began. One of these was the emergence from the research and development stages into reliable production of MOS/LSI (Metal Oxide Silicon/Large-Scale Integration) technology. This new technology increased equipment reliability and offered the possibility of much more sophisticated terminals at reasonable cost, owing to the very great compaction of components made possible. The second development was a direct consequence of the first: availability of the "smart" terminal containing a programmable processor.

The advent of programmable terminals was particularly significant for BALLOTS. A bibliographic application is quite different from the business applications for which hardwired terminals are usually designed. The programmable terminal offered the possibility of tailoring and optimizing the operational functions of the terminal for our particular application. Most important, the programmable terminal offered a sound solution to the overflow and field expansion problems discussed above.

The choice of terminal was narrowed to three units: the Spiras Systems' Irascope model TE (now designated model LTE), the Four Phase Systems' System IV/70, and the Sanders Associates' PDS (800 series).

The Spiras Irascope model TE had been developed in close collaboration with the Ohio College Library Center specifically for an on-line bibliographic system. Therefore the project gave this terminal special attention even though it was not a programmable unit. The review of this terminal produced much valuable information and many good ideas. However, the Irascope did not meet a number of project requirements. It was a highly specialized unit that would not have been of general use to the Stanford community. Although it offered block transmission, it did so only at 2,400 baud synchronous, rather than 9,600 baud asynchronous. Modifications to change the I/O would have raised the price significantly. And the fact that Spiras Systems was a small eastern manufacturer with few units in the western United States and no service organization of its own did not bode well for maintenance service.

The Four Phase IV/70 was attractive by virtue of its programmable processor and the fact that it is manufactured in the San Francisco Bay area. Although this might have been an acceptable unit for the BALLOTS application, it suffered from two limitations. First, it was able to display only 1,152 characters at one time. Second, it was available only in a clustered configuration. This meant that, although it was economically viable for a large user with many terminals, it was prohibitively expensive for the user with only one or two terminals. It was felt that this limitation would make the terminal impractical for potential CLAN members.

The final choice was the Sanders 800 series. It met all project specifications, and its programmable features appeared to make it the most flexible device for our application. Furthermore, it was easy to intermix clustered and stand-alone units, a consideration of economic significance for use in various-sized libraries. It was the only unit that accommodated data transmission via the same type of cable required for cable TV. (Because other projects at Stanford were interested in cable TV for transmitting data, it was considered highly desirable to have equipment accommodating a service that could later prove of importance to the library). Finally, at the time the comparative analysis was performed, the Sanders was the least expensive unit. Later, changes were made that increased the cost of the Sanders terminal, and its cost advantage over the Four Phase IV/70 evaporated. However, both these units were still less expensive than the Irascope TE. Finally, Sanders Associates was the largest and most experienced of the three manufacturers, and they had built up a service organization that was among the largest in the industry.

2.4.9 Real and Imaginary Terminals

In conclusion, there were two phases to the CRT screen design. First was the initial approach to the subject, which involved solving certain general problems. This phase can be conducted on paper with a hypothetical terminal. Second was the tailoring of the basic approach to a specific physical environment, to a specific terminal. As soon as a terminal had been selected, the criteria for designing screen formats could be totally formalized, a step that could not be taken beforehand. For example, different terminals not only have different character sets, they also have different numbers of characters on each line and different numbers of lines on the screen.

The preliminary design experience on the hypothetical terminal was most valuable. Had the terminal been selected before any design work had been done, the choice would have had to be made on much less specific and less mature requirements. And had the staff begun designing on a real terminal rather than a hypothetical one, the same learning experience would still have been required before it could be decided how to use that terminal. Actually, transfer of the design work done for the Sanders 720 to the Sanders 800 series was primarily a refinement. There were no basic changes. With the new device, some of the problems inherent in the earlier terminal could be solved relatively easily. No situation arose that required the design staff to backtrack and go off in a completely new direction.

CHAPTER 3

ANALYTIC STUDIES AND STANDARDS

This chapter presents several analytic studies made, standards created, and and decisions taken to support the design and implementation of BALLOTS II. The studies and standards incorporated in the text are as follows: Video Terminal Support for SPIRES and BALLOTS on Stanford's IBM 360/67; BALLOTS Data Element Notation; Video Terminal Evaluation; Video Terminal Screen Standard; BALLOTS Command Language; Coding and Description Standards for PL360; an ASIS 1971 Conference paper on on-line library network planning. In addition, cost studies conducted are described.

3.1 HOST COMPUTER ANALYSIS

A study was first made to determine the feasibility of a computer facility dedicated solely to library automation and other data file services; the idea was rejected because of the cost involved.

Since it was not found financially feasible for Stanford to dedicate a computer to library automation, an intensive study was made during the reporting period to determine the most appropriate available computer to support the BALLOTS production system. The two most promising computers were found to be the Administrative Computing Facility's IBM 360/40 and the Stanford Computation Center Campus Facility's IBM 360/67.

Two separate studies were conducted by the BALLOTS team with the programming teams from each facility. Preliminary designs were developed and an estimate made of the modifications and extensions of existing software required to support BALLOTS.

3.1.1 Administrative Computing Facility

The Administrative Computing Facility (ACF) was operating a 360/40 for use by the administrative sector of the University. This included jobs for the Registrar, the personnel and payroll departments, and the alumni records. A project was underway in the ACF programming area to design and implement an on-line administrative computing system using video terminals. The development activity had been about half completed at the time BALLOTS began investigating the use of the ACF computer. The video terminal used by the ACF development team was a Sanders 720, which had the hardwired capability of protected fields but not of field expansion. Since most of the administrative work could be carried out using the 720 CRT screen format with a file design that included a variable number of fixed-length data elements, there was no critical need for variable-length fields either on the CRT screens or in the file.

The BALLOTS programming team worked with the ACF programmers to determine the impact on both the ACF and the BALLOTS development schedules and intentions if the BALLOTS programming and production operations were carried out on the ACF computer. This study was conducted over a six-month period. One of the major portions of the study was an ACF review of the BALLOTS requirements, workloads, and development schedule, and a similar BALLOTS review of the ACF activities. After this was completed, the two teams produced a combined development task chart, describing over 250 tasks, that covered all of the current and planned activities for both ACF and BALLOTS. Merging the tasks on the same chart required an estimate of the amount of additional work that the ACF systems programmers would have to do to accommodate the BALLOTS requirements.

The study showed that the software changes necessary to support variable-length fields in file storage and on CRT terminals would require completely redesigning the ACF's file services software and also redesigning the terminal handler software. Changes to accommodate the additional number of terminals and the added transaction load would cause further software changes in the system. From the BALLOTS standpoint, more work would be required of programmers and analysts to produce a system of less benefit to the user than had been planned. Certain necessary aspects of the BALLOTS design--such as the overflow problem (see section 2.4.4)--were never satisfactorily accounted for in the ACF study. The cumulative impact of all these changes and compromises would have been a major change in the direction of BALLOTS development as well as in the schedule of the facility. It was therefore decided that Project BALLOTS should not be supported on the Administrative Computing Facility 360/40.

3.1.2 Campus Facility

During 1969 and early 1970, the Campus Facility machine was close to saturation. The installation software at that time was workable and efficient, but had not yet been fully optimized. Furthermore, there was a heavy batch workload. For this reason BALLOTS and SPIRES began by carefully investigating the ACF. But while the two projects were investigating the ACF environment, two changes took place in the Campus Facility: a gain in CPU cycle availability due to substantial software optimization, and a decrease in the overall workload. In the past year, reliability on the 360/67 has increased to the point where uptime is around 96 percent. Throughput in the high-speed batch partition has improved 40 percent. For example, the execution time for an average job has been reduced from 4.3 seconds to 2.2 seconds and the minimum job cost has been reduced from fifty cents to twenty-five cents. Text-editor (WYLBUR) throughput has increased 100 percent--i.e., it has doubled, effectively cutting costs to the user by 50 percent. These improvements to the operation of the 360/67 resulted in an average machine cycle availability of 30 percent.

Thus, the investigation of the Campus Facility showed that it could offer strong support, experience in developing time-sharing and virtual memory software, and on-line interactive systems that included text editing and compilaton. This software had been available for several years via typewriter terminals (primarily IBM 2741); 88 of 130 typewriter terminals on and off campus could be logged on at the same time. Software support for video terminals was not available, however, nor had a standard campus video terminal been selected.

3.1.3 Video Terminal Support for SPIRES and BALLOTS on the IBM 360/67

The preliminary study for developing and implementing BALLOTS at the Campus Facility showed that the additional transaction load from library automation could be absorbed without seriously affecting the existing software or service to users. The major addition required to the software would be support for CRT terminals. The Campus Facility provides services to faculty, students, and staff; and therefore must make all services as widely usable as possible. The CRT terminal chosen for library automation would also be available as a general campus CRT device, providing all the services of the Campus Facility. In addition to the powerful and flexible Sanders terminal chosen for BALLOTS, the Campus Facility elected to support a more modest (and much less expensive) CRT terminal, the Hazeltine 2000. The Campus Facility produced the following preliminary design report describing support for video terminals.

* * * DOCUMENT FOLLOWS * * *

1. BACKGROUND

Discussion began in January 1971 between the BALLOTS and SPIRES technical staff and the staff of the Stanford Computation Center Campus Facility (SCCCF) on CRT terminal support for the SPIRES/BALLOTS system. Several alternative designs were discussed. Each had its own effect on present SCCCF software, on core requirements, on overall system performance, on BALLOTS II development schedules, on reliability, and on cost.

2. ALTERNATIVES

Each alternative, with a discussion of its merits and disadvantages, is given below.

a. Via a 2701 into MILTEN

MILTEN, the communications subsystem, operates in an interrupt mode to support up to 88 IBM 2741 typewriter terminals. With this arrangement, a buffer pool consisting of 88 160-character buffers and associated control blocks must be reserved. The additional code and buffers to support thirty to forty CRT terminals (assuming 1,000-character screens), using either an interrupt or a polling discipline, would require a prohibitive amount of core. Such core could be taken from resident core in the nucleus, from execution pages within ORVYL, or from the large batch partition. All of these alternatives are considered unacceptable because of their effects on overall system performance. Furthermore, the MILTEN/WYLBUR and MILTEN/ORVYL interfaces now existing would be disturbed, thus requiring critical areas within the system to be redesigned and re-coded.

b. Via the existing PDP-9 into ORVYL

ORVYL, the time-sharing subsystem, now contains device-independent code that "talks" to a PDP-9 via the multiplexor channel and a 2701 Parallel Data Adapter. The PDP-9 contains device-dependent code for "foreign" computer links, paper tape devices, graphics scope devices, and teletypes. These are termed "external devices" and can be attached by a 2741 terminal program executing out of virtual memory under ORVYL. To augment the device-dependent code in the PDP-9 to support CRT terminals for BALLOTS II would either force a user to log on with a 2741 and switch to the CRT terminal (unacceptable from an economic standpoint), or force the BALLOTS II development staff to create a task-dispatching mechanism to sit underneath the one existing in ORVYL (thus seriously affecting the schedule). In addition, the reliability factor in the PDP-9 would be considerably degraded because of the unknown behavior of the various other external devices. It is also uncertain whether or not the PDP-9 would be capable of carrying the expected load.

c. Via a dedicated small computer into MILTEN and ORVYL

With a dedicated small computer providing buffering, line handling, error re-try, etc., it is considered feasible to provide a reliable interface between as many as 32 CRT terminals and the SCCCF 360/67 with a minimum of perturbation to existing system software. The CRT user may log on in line-by-line mode into MILTEN, with the small computer simulating the behavior of a 2741. After the user signifies that he wishes to be attached to ORVYL, a supervisor call is issued that commands MILTEN and the small computer to place the user in "full scope face" mode. From that point, all I/O takes place from pages of ORVYL execution space rather than from the MILTEN buffer pool. The control block in the buffer pool merely has a flag set, with a pointer to the real buffer in the ORVYL partition. Such an approach is expected to require minimal core on the 360/67 and feasible changes to the existing system software. It was therefore chosen as the basis for CRT support on the SCCCF 360/67.

3. CHOICE OF SMALL COMPUTER

After conducting a survey, the Small Systems Group at SCC reduced the set of possible small computers to Data General's Super Nova, Varian Data Machines' 620i, and Digital Equipment Corporation's PDP-11. From the standpoints of reliability, flexibility, and operating characteristics, the PDP-11 stood out as the best alternative, especially with regard to its flexible busing scheme. Furthermore, local expertise has been built up during the installation of a front-end PDP-11 at the ACME (Medical School) Facility of SCC.

4. TERMINAL COMMUNICATION

There are three alternatives for arranging communication between the terminals and the PDP-11. They are: (1) dedicated lines, (2) multidropped lines, (3) time division multiplexing (TDM). In comparing the first two alternatives, it was felt that dedicated lines would increase the user cost beyond practical limits. The second alternative--multidropped lines--was recommended as an interim choice, pending completion of a study on the feasibility of TDM. It was not felt that the alternative selected would lead to the scrapping of any significant amounts of either hardware or software should TDM become the accepted approach. There was the further advantage of allowing parts of the project unaffected by this decision to commence immediately. The remainder of this memorandum therefore addresses itself to a multidropped system.

5. CHOICE OF TERMINALS

Two terminals have been chosen as standard CRT terminals for front-end support for the Stanford community and all its users, including off-campus network members. They are the Hazeltine 2000, which has been used with success in the Campus Facility, and the Sanders Associates 800-series programmable terminal. (See sections 2.4.8 and 3.2 for the analysis of CRT terminals that led to Sanders as the choice.)

6. IMPLEMENTATION SCHEDULE

The schedule for implementing the front-end system has been divided into two stages:

Stage 1: Deliver and install all equipment in Pine Hall, and when sufficient progress has been ensured, three Sanders display terminals will be installed at the Main Library. It is anticipated that programmable terminals will be multidropped to reduce communication costs.

If the DEC multiplexor interface becomes available, the PDP-11/2701 interface and the PDA itself will be replaced by a DEC-DX11 interface. This will replace a 2701 that has a rental cost of $12,000 based on a forty-month period.

Stage 2: Deliver and install at least one inexpensive CRT terminal and integrate its operation into existing Stage 1 support. This will include but will not be limited to operation on the public switched telephone network as well as on dedicated circuits. The inexpensive terminals will not be multidropped. Some economies may be achieved in the hardware interfaces to the PDP-11 if a transmission rate less than 9,600 bps is selected. This area remains to be elaborated.

7. BILL OF MATERIALS

A. DEC

1	PDP-11/20-CA computer with 4k memory	$11,450
2	MM11-E 4k 16-bit 1.2 us add-on memory	7,000
1	KW11-A real-time clock	250

1	DD11-A mtg. panel	175
1	DX-11 MPX channel interface	5,000 est.
2	KL-11X full duplex line interfaces	800
	Total	$24,675

B. Engineering

1. PDP-11 IBM 2701 PDA interface

1 DR11	$ 400
Logic	400
Miscellaneous	300
Subtotal materials	$ 1,100
Labor (seven man-days)	700
Total	$ 1,800

2. Multidrop interface boxes

Receiver/drivers	$ 180
Miscellaneous	120
Subtotal materials	$ 300
Labor (one man-day)	100
Total	$ 400
times four boxes	x4
Total	$ 1,600
plus extra receiver/driver on KL-11	180
Total	$ 1,780

Note: Each multidrop interface box is capable of driving four terminals within one hundred feet of each other. If terminals could be thus clustered in the library, the cost for multidrop interfaces would be $400.

8. SITE FOR THE PDP-11

The Operations Manager, Campus Facility, sees no problem in locating the PDP-11 near the current PDP-9 installation in Pine Hall.

* * * DOCUMENT ENDS * * *

Based on the results of this study, a formal decision to implement library automation on the Campus Facility was made and design and programming continued. The next step for the Campus Facility included the specification for ORVYL support of CRT terminals, the ordering of the PDP-11, and the building of the interface hardware. It was found that all the deficiencies in the plan to implement at the Adminstrative Computing Facility were overcome or could be overcome at the Campus Facility.

3.2 VIDEO TERMINAL EVALUATION

NOTE: This analysis was based on information about products and prices available as of May 1, 1971. It does not reflect the changes in products and costs and the introduction of new products that have occurred since that date.

* * * DOCUMENT FOLLOWS * * *

In February 1971, Project BALLOTS began an intensive search for an acceptable CRT terminal to be used in an on-line bibliographic application. Requirements (Table I) were based on the conceptions of the BALLOTS staff, experiences of similar projects, and the hardware and communications environment of the Stanford Computation Center (SCC). The survey of available terminals began with a review of technical publications such as the Auerbach Data Communications reports. also reviewed were articles and advertisements in data processing and library trade journals. On the basis of the information available at that time, specific data about the terminals were gathered in Table II.

A preliminary comparison of the terminals (Table II) against our requirements (Table I) pointed to six potentially acceptable terminals. The most difficult requirement to meet was the one for protected format with expandable input fields. The six terminals, and the modifications necessary to make them meet the formatting requirements, are listed in Table III.

After closer examination of the units, this group of six was narrowed to a group of three: the Four Phase IV/70, Sanders 804, and Spiras Irascope TE. The Datapoint 2200 was rejected because it had only a 12-line by 80-column screen (960 characters) and because it did not have adequate keys for cursor control. The Delta TelTerm II was rejected because the use of an external minicomputer to supplement the terminal's functions was not considered desirable while there were questions about the reliability of the computer-terminal combination and the fiscal strength of the company. The Imlac PDS-1 was rejected because of flicker in the visual image, especially when more than 1,200 characters are displayed, and because of higher unit cost.

The three remaining terminals were compared by function (Table IV). Differences in memory size and character set were

TABLE I

BASIC TERMINAL REQUIREMENTS

DISPLAY:	Minimum 12" diagonal CRT Minimum 1000 character display Minimum 24 line by 48 column display Upper/lower case character set
KEYBOARD:	Typewriter layout (ANSI x 9A9/199B preferred)
FUNCTIONS:	Protected format with expandable input fields Full cursor control left right up down tab home Full editing capabilities character insert character delete character overstrike line insert line delete
COMMUNICATIONS:	9600 baud (Asynchronous preferred) Block I/O Stand alone capabilities

TABLE II

TERMINALS CONSIDERED

MANUFACTURER	MODEL
Beehive	I, II, III
Courier	
CTC	Datapoint 2200 Datapoint 3300
DEC	VT05, VT06
Delta Data	Tel Term II
Hazeltine	2000
IBM	2260, 3270
ICL	7181
Imlac	PDS-1
Infoton	Vista
Lear Siegler	7700
Sanders Associates	720, 800 series
Spiras Systems	Irascope TE
Uni Comp	522
XDS	
Four Phase	IV/70

TABLE III

Terminal	Necessary modifications to meet format requirement.
CTC Datapoint 2200	Add to manufacturer's software for terminal's processor.
Delta Data System Tel Term II	Run terminal in full duplex with external minicomputer supplementing terminal functions.
Four Phase IV/70	Add to manufacture's software for terminal's processor.
Imlac PDS-1	Add to manufacturer's software for terminal's processor.
Sanders 800 series	Add to manufacturer's software for terminal's processor.
Spiras Irascope TE	Specify reprogramming of certain ROM's by manufacturer.

TABLE IV

COMPARISON OF TERMINAL CHARACTERISTICS

	FOUR PHASE IV/70	SANDERS 804	SPIRAS IRASCOPE TE
PROCESSOR	+ Programmable MOS/LSI 1.9 Microseconds	+ Programmable MOS/LSI 6.6 Microseconds	- Hardwired MOS - one time programmable ROM 350 Nanoseconds
MEMORY SIZE	0 For 8 terminal cluster: 12-24k bytes processor and display	0 For each terminal: 2.5-8k bytes processor 1-2k bytes display	0 For each terminal: .5k bytes processor 1-4k bytes display
DISPLAY SIZE	- 1152 characters 24 lines x 48 columns	+ 1920 characters 24 lines x 30 columns	0 1152 characters 32 lines x 80 columns
CHARACTER SET	0 120 up/low	0 96 up/low	0 124 up/low and diacritics
CONFIGURATIONS	+ Clusters of 1-8, opt. disk (stand alone available June 72)	+ Stand alone (clusters of 1-8 x 4 available May 72)	- Stand alone
COMMUNICATIONS	+ To 9600 baud asynchronous (higher rates achievable)	+ To 9600 baud asynchronous, higher rates for clusters, compatable with cable TV	- To 2800 baud asynchronous (9600 baud asynchronous available March 72 with engineering charge & added unit cost)

TABLE IV (continued)

COMPARISON OF TERMINAL CHARACTERISTICS

	FOUR PHASE IV/70	SANDERS 804	SPIRAS IRASCOPE TE
SERVICE	+ By manufacturer, local firm	+ By manufacturer, largest terminal service organization in the industry	- By third party under contract to manufacturer, no arrangement in Bay Area at present
RELIABILITY	0 Unknown, units in field two months and operating well Firm's first and only product	0 Unknown, no units in field Previous products acceptable	0 Unknown, one prototype unit delivered Previous terminal withdrawn
	+3	+5	-4

not felt to be significant for our application. Reliability is an unknown factor for all three units. The flexibility of the programmable processor and the flexibility of units providing both stand-alone and clustered configurations made the Four Phase and Sanders terminals more attractive than the Spiras unit. The Spiras unit was also less desirable than the other terminals because of its synchronous transmission and late delivery of 9,600 baud. The Sanders terminal is preferable to the Four Phase on the basis of display size. And the Sanders is built by a well-known, established company.

Table V contains a cost comparison of the three terminals. No information is available on a cost of a Four Phase stand-alone unit. The only figure, quoted tentatively, for the Sanders cluster was $27,200 for a cluster of eight. Other Sanders prices are preliminary. Given these limitations, an evaluation of the data in Table V indicates that: (1) for quantities of one to three terminals, a stand-alone unit is definitely desirable; (2) the Spiras unit is significantly more expensive than the other two; (3) the Sanders 800 series appears to be slightly less expensive than the Four Phase; and (4) the Sanders 804 is the least expensive alternative for our first installation of four or five units. And for our first installation, the Sanders 804 stand-alone also provides more flexibility than the Four Phase IV/70, which is currently available in a cluster only.

In addition, the Sanders terminals are engineered to receive cable TV signals form coaxial cable. This fact makes these terminals more desirable for future campus communication developments at Stanford University, and could be an important factor in future research work.

A comparison of delivery schedules (Table VI) shows the Sanders to be somewhat later than the others. However, the delivery constraint does not seem to be severe enough to outweigh the other advantages of this unit.

We therefore recommend the Sanders 800 series CRT terminals for use in this application.

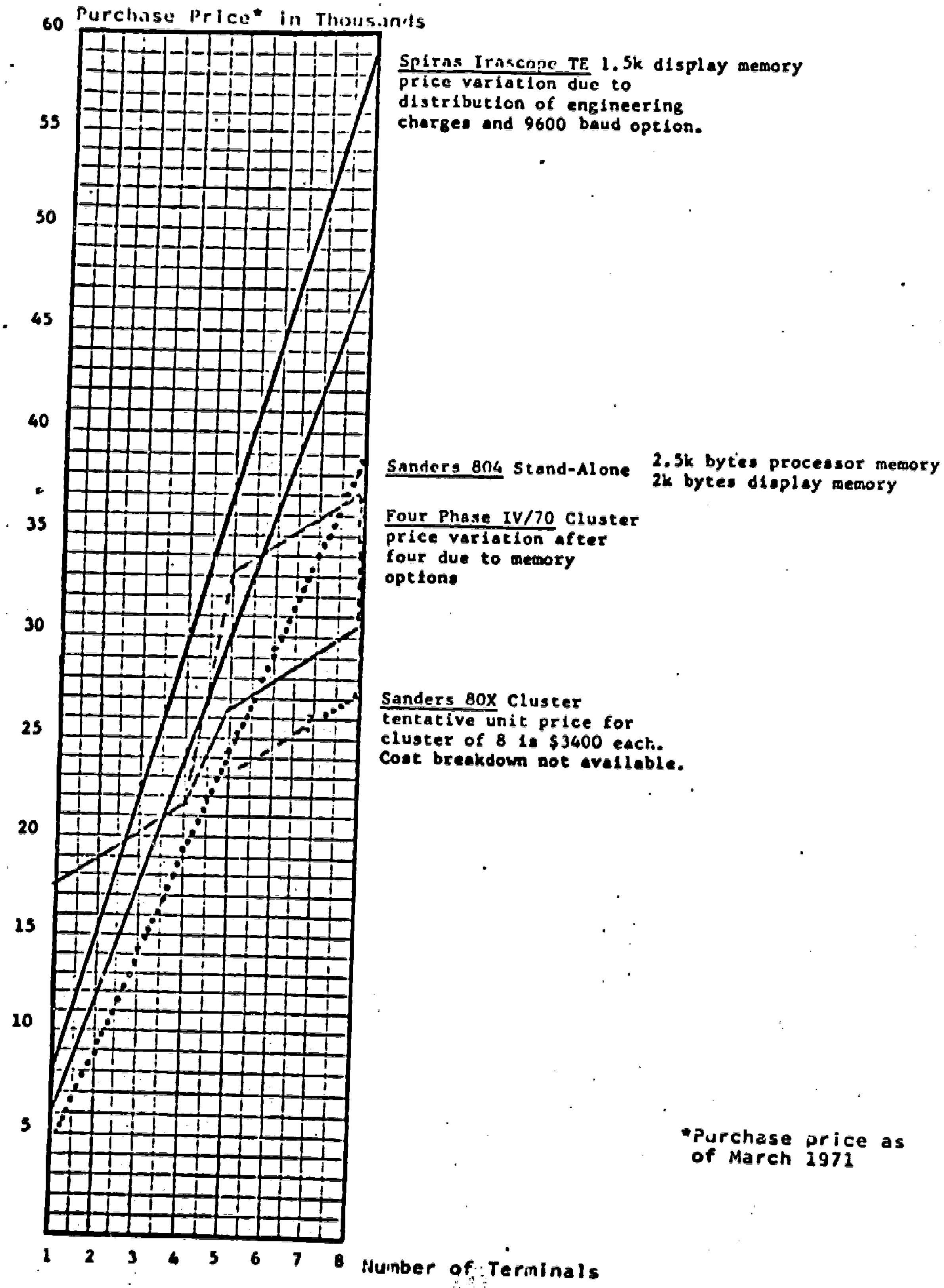
* * * DOCUMENT ENDS * * *

3.3 VIDEO TERMINAL SCREEN STANDARD

Once the CRT terminal evaluation had been completed and the Sanders 804 terminal had been chosen, the characteristics of the terminal as they affected the user, the programmer, and the system analysts were documented; from this information a standard was developed. This standard treats (1) the method for line overflow when a data element takes up more space than has been

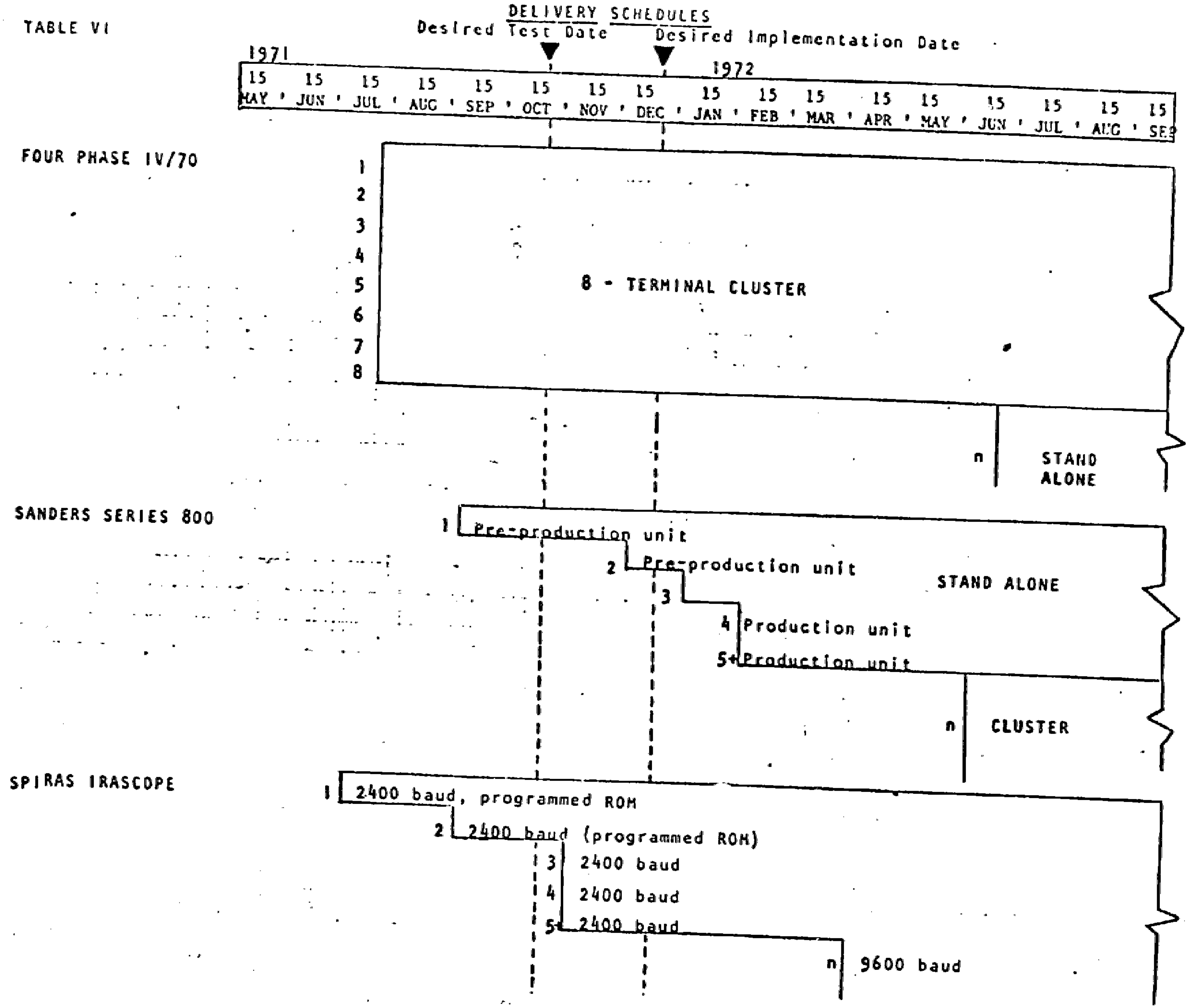
TABLE V

COST COMPARISON



*Purchase price as of March 1971

TABLE VI


DELIVERY SCHEDULES
Desired Test Date
Desired Implementation Date
1971
1972
15 MAY
15 JUN
15 JUL
15 AUG
15 SEP
15 OCT
15 NOV
15 DEC
15 JAN
15 FEB
15 MAR
15 APR
15 MAY
15 JUN
15 JUL
15 AUG
15 SEP
FOUR PHASE IV/70
1
2
3
4
5
6
7
8
8 - TERMINAL CLUSTER
n
STAND ALONE
SANDERS SERIES 800
1 Pre-production unit
2 Pre-production unit
3
4 Production unit
5+ Production unit
STAND ALONE
n
CLUSTER
SPIRAS IRASCOPE
1 2400 baud, programmed ROM
2 2400 baud (programmed ROM)
3 2400 baud
4 2400 baud
5+ 2400 baud
n 9600 baud

allocated on the CRT screen; (2) the location and definition of the command line; and (3) the definition and location of error messages.

* * * DOCUMENT FOLLOWS * * *

Definition and Use of Protected Fields

INTRODUCTION

This is a description of the principles of interaction between the library user and on-line portions of the BALLOTS-MARC modules. A program called the BALLOTS subprocessor (BSP) will control the input, editing, storage, retrieval, and display of data. The user will communicate with the system through a Sanders 804 (stand-alone) or 810 (clustered) programmable CRT terminal. The input and display of data will be organized within formats for ease of use and verification. The operator gives directions to the BSP through a command language. These elements will be discussed below.

SYSTEM MODEL

The following model is provided as a point of reference for the terms and explanations that follow. This is a conceptual model, not a program design. The BALLOTS subprocessor may indeed be organized in a vastly different manner, but the user will be able to conceive of operations according to this model. The primary purpose of the model is help illustrate what the elements of the system are supposed to do. How the system actually effects the actions described is entirely up to the programmer.

FORMATS AND FRAMES

A format, or format routine, is a program that serves as an intermediary between the user and the data base. It organizes the interchange of data into logical units and allows the user to deal with logical subsets of the data elements that make up a single record. The format aids the user by labeling data elements and displaying them in a consistent manner. The format also edits for accuracy and completeness the data input by the user.

A format consists of a set of format and display rules, and a set of edit and storage rules. The first set of rules is used when data is output to the terminal from the computer. The second set is used when data is input from the terminal to the computer.

What the operator actually sees on the face of the CRT is called a frame. A frame is created by the format routine. A frame always consists of 24 lines of 80 characters each. The format routine takes data from the data base, organizes and labels it according to the formatting and display rules, and sends it, along with positioning and protection information, to the terminal where it is stored in the display memory. The terminal uses the positioning and protection information to write the frame on the face of the CRT.

A format routine may create one or more frames before it completes its processing. And the same routine will create a different number of frames in different circumstances. This is caused by the fact that although a specific format always deals with the same group of data elements, the length of the data elements may vary widely from one occurrence to another.

FIELD EXPANSION

One of the most important characteristics of BALLOTS CRT screen formats is the ability to have input fields of flexible length. This function is carried out by the terminal processor without interrupting the remote CPU. If an input field extends into position 80, that field may expand in increments of 80-position lines, one at a time. After position 80 is filled, if the next character is not a tab or carriage return, which would take the cursor to the next field, the terminal program will insert another input line for the current field. This line will be inserted immediately below the current line, and succeeding lines in the frame will be pushed down one line. The last line in the frame will be pushed off the bottom of the screen and will be lost from the display memory. But the information displayed in this last line is not lost from the data area of "user scratchpad memory" (USM). When the frame is transmitted, the format routine will rewrite the record in the data area to accommodate the inserted line(s). The format routine will then build another frame and send it to the terminal. The beginning of this next frame will overlap the last two lines of the previous frame.

If the operator has input data in the last line and then pushes that line off the bottom of the frame, the input data will be lost because it has not been transmitted to the system.

PROTECTION

It is possible to code characters in the display memory so that they cannot be altered from the terminal keyboard. A format must send information specifying that certain characters are to be protected. In BALLOTS, all positions that are not to be made available to the operator for input are designated as protected.

When making up a frame from a format, all 80 positions of each of the 24 lines must be accounted for.

DESCRIBING FORMATS

To completely define the requirements for a CRT screen format two things are required. The first is a visual model of the display of data on the CRT screen. This is done on a BALLOTS grid form (a sample of this form is included at the end of Appendix B), using as many lines as necessary to accommodate the group of data elements in the format. Division and overlap between frames is a flexible matter because of possible field expansion.

The second thing necessary to define a format is a set of processing rules. These rules specify the relationships between, and the rules governing, the data elements included in the format. These rules may also specify special considerations to be observed in dividing a record into frames.

GENERAL CONSIDERATIONS OF CRT SCREEN FORMAT DESIGN

1. All frames created by format routines (except search formats) will consist of two parts, the header and the body. Search frames contain only a header.

Header

2. The header occupies the first three lines of all frames except search frames. The header occupies all 24 lines of a search frame. When a record must be divided into more than one frame, the header will be repeated at the beginning of of each frame.

Control Field

3. The first element of the header is the control field. This line contains the Format I.D. in columns 6-11; File I.D., columns 17-28; Record I.D., columns 34-42; Function I.D., columns 48-55; Library I.D., columns 61-67; and operator I.D., columns 73-76. Values for these items are always left-justified. An explanation of each of these items can be found in the BALLOTS Data Elements Notebook <7>.

Message Field

4. The message field is the second element of the header. It occupies line 2, positions 1-80. It is protected. It is used to display diagnostics

concerning the command field and general messages from the system. If a message exceeds 80 characters, it will expand and push down succeeding lines. In a search frame the message field may be up to 22 lines in length.

Command Field

5. The command field is the third element of the header. The command field is an input field, i.e., it is unprotected. It follows the message field in all frames but search frames. The command field occupies positions 1-79 of line 3. It does not extend into position 80 because the command field is not allowed to expand and push down succeeding lines in non-search frames. In search frames the command field begins in position 1 of the line immediately following the message field and continues to the end of the frame. The command field of a search frame may be up to 22 lines in length.

Body

6. There are two types of bodies: (1) display bodies and (2) input bodies. Update bodies are a form of input bodies.

Display Bodies

7. In display bodies, all lines are protected. The operator may input in the command field only.

8. Common, easily identified data elements need not be labeled in a display body.

9. Rare data elements, and/or data elements whose identity is not obvious from their value, must be labeled.

10. In display bodies, textual data should be organized into logical paragraphs.

11. Short, fixed-length data elements may be organized into multiple columns if desired. Columnar alignment should be made on the first character of the data element value. All lines containing more than one data element should conform to the same columnar structure.

12. Columnar data should be grouped together, not mixed with paragraphed data.

13. When the same data element, or group of data elements, is included in several different formats, the display position and organization of such elements should be consistent across formats.

Input Bodies

14. Input bodies may contain two types of entries: (1) data element entries, and (2) instruction entries.

Data Element Entries

15. Each data element entry contains two parts: (1) a label, and (2) an input field. There are two types of labels and two types of input fields.

Labels

16. All labels are eight characters long. The first two character positions are reserved for the display of error codes. The third position is blank. The fourth through seventh positions are for the data element mnemonic. The eighth position is blank.

17. The first type of label is a "prompted mnemonic" label. In this type the format routine provides the label. It is right-justified to position 7 of the label, and cannot be altered by the operator. All eight positions in this type of label are protected for strict input formats. In update formats, the eighth position is unprotected. The operator may insert an asterisk in this eighth position to signify that the element is being updated.

18. The second type of label is an "operator supplied mnemonic" label. In this type, positions 4-7 of the label are unprotected, and the operator inputs a mnemonic anywhere in these positions. Positions 1-3 and 8 are protected. In update formats (see 17) position 3 is unprotected to allow the operator to insert an asterisk to signify an update.

19. Entries for data elements that may be multiple within a structure (BIBS, AS, IS) must use "operator-supplied mnemonic" labels. If the structure is multiple but the data element is singular within the structure, e.g., MDX in AS, or LOC in IS, prompted mnemonics should be used. The program will know how many copies of

the structure are needed to present a given record in a given format.

20. The two types of labels should not appear in the same format.

Input Fields

21. The two types of input fields are (1) fixed length fields of less than 72 characters, and (2) expandable fields.

Fixed-Length Input Fields

22. Fixed-length input fields have a maximum length of 71 characters. Since an input field must be preceded by an eight-character label, an input field longer than 71 characters would extend into position 80. Only expandable input fields may use position 80.

23. If a fixed-length input field is short enough, it is possible to include more than one data element entry in a single line. This is permissible if (1) there are at least five spaces between the end of each input field and the beginning of the next label, and (2) if all lines in a format containing multiple data elements conform to the same columnar structure.

24. Columnar lines should be grouped together, not mixed with single data element lines.

Expandable Input Fields

25. For a field to be expandable, it must begin in position 9 (immediately after the label that began in position 1) and must extend into position 80. If the 73rd character input by the operator (the character that would occupy position 81) is not a tab or carriage return, the terminal program will insert another input line (see FIELD EXPANSION).

Instruction Entries

26. Instructions are defined as data that will be processed by a program and then discarded. Data element values, on the other hand, become part of the data base and are stored. An example of an instruction would be information

telling the program to print out a set of catalog cards.

27. Instruction entries begin with a two-position error code field. This is followed by a single space. These first three positions are protected. The input field for the instruction follows. It is fixed in length and is not labeled.

28. Instruction fields are prompted with underscore characters. This differentiates them from data element input fields.

29. Instructions should be in the form of short codes, such as "M" for "material received" or "I" for "invoice received" or "MI" for "material received with invoice."

30. The length of an instruction input field is determined by the length of the longest instruction, or set of instructions, that may occupy that field.

31. Instruction entries must begin in position 1.

* * * DOCUMENT ENDS * * *

3.4 BALLOTS COMMAND LANGUAGE

The following is a description of the BALLOTS command language involved in the BALLOTS-MARC module. (See also section 2.4.7.) As other modules are added to the BALLOTS system, the command language will expand. BALLOTS commands should not conflict with other command languages with which users may be familiar. The same function should have the same command word in all systems, and the same command word should call the same function in all systems. The main difference between BALLOTS and the other existing command languages at Stanford is that BALLOTS deals with data one frame at a time rather than one line at a time.

* * * DOCUMENT FOLLOWS * * *

BALLOTS-MARC COMMANDS

CANcel
CONTinue
DISplay
ENTer

FINd
- AND
- NOT
- OR
- BACkup

format routine calls:

BF01	Search Results - Full Bibliographic
BI01	Input - Bibliographic
BI02	Input - Additional Bibliographic
GS01	General System Screen
HH01	Input - Holdings
OR01	Input - Ordering
SI01	Search Input
SI02	Search Continuation

IGNore
KEEp
LOGON
LOGOFF
paging
- +
- +B
- -
- -B

REStore
SCRatch
SEArch
SET
SKIp

Most commands may be input using either the first three letters or the whole command. The only exceptions are LOGON and LOGOFF, which cannot be abbreviated. All commands may be input in upper or lower case.

In the discussion of commands, reference will be made to the following levels of activities. Each level may be considered a logical set made up entirely of one or more occurrences of a member or members of the next lower level.

FUNCTION	Ordering Function	
	Catalog Function	
	Scratch Function	
PROTOCOL	Search Protocol	
		Standing Search
	Build Protocols	Order
		Catalog
	Scratch Protocol	
FORMAT ROUTINE	BF01	
	BI01	

	BI02 GS01 HH01 OR01 SI01 SI02
CYCLE	Input Cycle Overflow Cycle Error Cycle
STEP	Output Step (output of a frame by the system) Input Step (input of a frame by the operator)

Build protocols are those that write to the data base. They are called build protocols because they build a record in the data area of "user scratchpad memory" (USM). They are more strictly controlled than are protocols that only read from the data base.

A format routine, or simply a format, will often consist of a single cycle. If a format overflows a single frame, an overflow cycle will be initiated. If the edit program finds an error in incoming data, an error cycle will be initiated.

A cycle is the period beginning after the operator has transmitted a frame and continuing until the operator transmits the next frame.

CANcel Command

CANcel is one of the two commands that can end a build protocol. The other is the ENTer command. It is impossible for the operator to exit from a build protocol without using one of these commands. The effect of the CANcel command is to clear the build area in USM. When the build area is cleared by a CANcel command, the status of all data is reset to what it was at the beginning of the build protocol: no files are updated, no print records are produced.

CONTinue Command

In BALLOTS-MARC the CONTinue command will be used to call for more input positions in a BI02 format when these additional positions are not provided by automatic paging. This condition will occur when the following three factors are all present: (1) all data in USM that should be included in a BI02 format has already been displayed on previous frames of BI02 or is being displayed on previous frames of BI02; (2) the last data element on the screen is in line 24 but does not continue past position 77; (3) the operator has more data elements to input in the BI02 format. The same is true for the HH01 format.

DISplay Command

The DISplay command will cause the records pointed to by the current results stack of the search area to be displayed in the BF01 format. The records will be displayed in CRD order. The display command does not end a search protocol, and it does not affect the results stacks or the build area in USM.

ENTer Command

ENTer is one of the two commands that can end a build protocol. The other is the CANcel command. It is impossible for the operator to exit from a build protocol without using one of these commands. The effect of the ENTer command is to take the record in the build area in USM and update files and/or produce print records as necessary. The successful execution of the ENTer command signifies the acceptance of the new record, or update, into the system. Finally, ENTer releases the build area.

FINd Command

The FINd command initiates a search protocol. It will begin by clearing the search area in USM. The actual search will be handled through the SPIRES/BALLOTS common software. The AND, NOT, OR, and BACkup commands may occur within a FINd command.

A search request is composed of the command "FINd" followed by the name of at least one index and at least one value for that index. The indexes available in BALLOTS-MARC will be personal name (PN), corporate-conference name (CN), title word (T), and LC card number (CRD). If more than one index-value pair is contained in a request, each pair must be connected with one of the logical operators "AND," "OR," "NOT." A request containing only one pair is called a simple request. A request containing more than one pair is called a compound request.

Examples:

a. (simple) FIND TITLE HONOR
b. (simple) FIND T GLORY
c. (compound) FIND PN SMITH AND JONES
d. (compound) FIND PN SMITH AND JONES AND T HONOR

In example c, the system would find all works written both by someone named Smith and someone named Jones. In example d, the system would find all the works that were written both by someone named Smith and someone named Jones and that had the word "Honor" in their titles.

The precedence of logical operators is left to right, with grouping capabilities provided by the use of parentheses. A new search is started by specifying the command verb "FINd."

Example: FIND PN HUNTER AND (T PRAY OR T REWARD)

If several values are specified for one index in a request, the name of the index need not be repeated.

Example: FIND T HAPPY OR GLEEFUL OR ECSTATIC

If several title words are specified without logical operators connecting them, "AND" will be assumed and need not be specified.

Example: FIND T TAMING OF THE SHREW

The system responds to a FINd command with a count of the number of records that match the specified criteria. Such a response always appears in a SI02 format. If the number of matching records is exactly one, that matching record will automatically be displayed on a display screen (always BF01 in BALLOTS-MARC) and there will be no statement of the search results. If the number of matching records is not exactly one, the number of records found in the search will be displayed on the SI02 screen and the user can enter a request that will increase or decrease the number of records found.

The user may go through many search iterations, entering successive requests to reduce or expand the set of records that meet his specifications. Occasionally, he may enter a request that reduces or expands the search result by much more than is desirable at that point. Rather than have to re-enter the entire search specifications, the user may then use the BACkup command. This command allows the user to request that the search status be reset to what it was before the last request. If a user enters a request that reduces the set of records found to zero, the system automatically backs up to its status before that request.

Special conventions are involved in specifying index values for the personal name index. Remember that the user may input in upper or lower case without having any effect on the search.

Format Routine Call Command

The format routine call command consists of a format I.D. This command is used to call up a certain format, subject to the rules of the protocol. In some cases this command will initate a protocol. This command can be used to jump ahead in a protocol, bypassing intermediate screens.

IGNore Command

The IGNore command instructs the system to disregard all actions taken during the current cycle. The contents of USM will in no way be affected by the transmission of the current frame. The system will respond with the transmission of a frame identical to the last frame transmitted. If that frame was blank, the retransmitted frame will be blank, regardless of how much data the operator may have input before issuing the IGNore command. This equals a "clear variable data" function within the terminal. If the last frame contained data from USM, the retransmitted frame will again contain that data unaltered. If the IGNore command is linked to another command, e.g., IGN/ENT, the retransmission of the frame can be circumvented.

KEEp Command

The KEEp command initiates a standing search build protocol. The effect of this command is to take the current form of the FINd command from the search area, pass it through the standing search edit rules, and display the edited search command for review by the user. The KEEp command builds a standing search record in the USM build area at the same time. The KEEp command does not affect the search area and the result stacks contained therein.

Paging Commands (Display Only)

The four paging commands are +, +B, -, and -B. In the MARC system they are used only with the BF01 format. The + command is used to advance to the next frame when a single record is too long to be displayed in one frame. If a record is only one frame long, or if the last frame for that record is currently being displayed, the + command will be rendered inoperable. The + command will not page ahead to a frame displaying another record; the +B command must be used for that.

The +B command is used to page ahead to the next bibliographic structure. That is, it will page ahead to the first frame of the next record. It will jump over any remaining undisplayed frames of the current record and go directly to the beginning of the next record. If there are no succeeding records, the +B command will be rendered inoperable.

The - command is the opposite of the + command. It is used to back up to the previous frame of a record that is too long to be displayed in one frame. Obviously, the operator must have first issued a + command in order to be in a position to enter a - command. If a record is only one frame long, or if the first frame of a multiframe record is currently being displayed, the -

command will be rendered inoperable. The - command will not page back to a frame displaying another record; the -B command must be used for that.

The -B command is used to page back to the previous bibliographic structure. That is, it will page back to the first frame of the previous record. It will jump over any previous frames for the current record, and will also jump over any trailing frames for the previous record. If there are no previous records, the -B command will be rendered inoperable.

Automatic Paging

Frame overflow in input and update format routines will cause automatic forward paging. In a prompted mnemonic format, e.g. BI01, the format routine has a list of all data elements contained in the format. If any of these elements has been pushed off of a frame, the routine will automatically transmit another frame. This succeeding frame will repeat the last two lines of the preceding frame and follow these two lines with the rest of the data elements in the format. In operator-supplied mnemonic formats, e.g. BI02, if there is any data beyond position 77 of line 24, the routine will automatically transmit another frame with a two-line overlap of the preceding frame. In the case of an operator-supplied mnemonic update format, if there is data in USM that has not yet been displayed, this data will cause transmission of another frame regardless of the contents of line 24.

If the operator finishes a data element in line 24 before reaching position 78 and wishes to input additional data elements, the CONTinue command must be used.

REStore Command

The REStore command re-initiates a format routine. If the REStore command is issued without a modifying format I.D. following it, it is assumed to refer to the format currently displayed. A REStore command always resets the format routine to the first frame. The REStore command is executed after the current frame has been processed. (Compare with IGNore command.)

If the REStore command is followed by a format I.D., the current frame at the terminal will be processed and then the named format will be displayed with data from USM. This allows the operator to back up in a build protocol. The format I.D. must refer to a preceding format routine within the current protocol. If the format I.D. refers to a succeeding routine, or to a routine in another protocol, the REStore command will not execute.

The second form of the REStore command may perhaps not be implemented as a part of the BALLOTS-MARC module.

RES BI01 returns to the first frame of the BI01 format for the record being constructed in the build area.

RES BI02 will return to the first frame of the BI02 format for the record being constructed in the build area.

RES HH01 returns to the first frame of the HH01 format for the record being constructed in the build area.

RES OR01 will return to the first frame of the OR01 format for the record being constructed in the build area.

SCRatch Command

The SCRatch command is used to delete a record. In the MARC system, the SCRatch command will be used to delete standing search requests. For this use, the command will have the the following form in the BDEN notation (see section 3.5):

SCR(atch)<SP(1)><SID>

where <SID> is a standing search I.D. number.

SEArch Command

The SEArch command in the MARC module will initiate a search protocol and prompt an SI01 format. In later modules this command may be used to specify which of several files are to be searched. In the MARC module, BMRC is the default, one-and-only file.

SET Command

SET FUN(CTION)=<function name>
SET LID=<library I.D.>
SET OID=<operator I.D.>

The SET function is used to change the last three fields of the control line. The initial value of these fields is set during the LOGON procedure.

SKip command

The SKip command is used to jump over succeeding frames of a format to another format. If no format I.D. is specified, the

SKIp will be to the next format in the protocol. The SKIp command may be followed by a format routine call command to skip over succeeding frames and intermediate formats in a single bound. (See linking of commands.) The SKIp command is executed after the data in the current frame has been processed. The SKIp command in no way affects the contents of USM.

When in BALLOTS-MARC the operator enters in build order or build catalog record protocol, USM will contain the full MARC record retrieved by the search. The use of SKIp to by-pass any part of BI01 or BI02 will not delete the portion of the MARC record that would have been prompted in the skipped part.

Linking of Commands

In some cases it is possible to link commands together. This is particularly true at the end of a protocol where an ENTer or CANcel command may be linked to another command to initiate a new protocol. Linking is done by using a slash, "/."

CAN/DIS
CAN/FIN <search argument>
CAN/LOGOFF
CAN/SEA
CAN/BF01
CAN/GS01
CAN/SI02
CAN/SI01

ENT/DIS
ENT/FIN <search argument>
ENT/LOGOFF
ENT/SEA
ENT/BF01
ENT/GS01
ENT/SI01
ENT/SI02

IGN/RES (format I.D.)

SKI/ENT
SKI/HH01
SKI/OR01

Procedure of Commands

All commands except CANcel and IGNore are processed after the data in the current frame has been processed. If an error condition exists, the current frame will be redisplayed and the operator must then correct the errors. If the command was originally prompted as part of the format, it will be reprompted when the frame is redisplayed; otherwise, the operator must reissue the command. In the case of CANcel and IGNore, the command is executed immediately; the data in the current frame is not processed and there are no error cycles. If a command is invalid, a diagnostic message will be displayed in the message line.

* * * DOCUMENT ENDS * * *

3.5 BALLOTS DATA ELEMENT NOTATION

BALLOTS development depends in part on the facility with which data element descriptions and processing rules can be formulated and communicated. It was found both inefficient and ambiguous to use conventional, prose descriptions for this purpose. Consequently, a method was developed that allows very precise, shorthand documentation of this information. This method is called BDEN--BALLOTS Data Element Notation. BDEN is somewhat similar to BNF (Backus-Naur Form), a familiar metalanguage. A BNF analyzer was used to detect any ambiguities that might arise in data element descriptions written in BDEN. BDEN was developed to overcome three deficiencies in conventional BNF: (1) five to six different productions were required to specify and describe a data element, (2) video terminal screen column positions could not be specified for input work; (3) it was difficult to show which components of a complex data element belonged together. BDEN enables a data element to be completely specified with a single production, in a single line. In addition to describing the data elements, BDEN is also used to describe output formats and the location of data elements. The language has proved a useful means of ensuring accurate communication between the library analysts and the programming staff. Members of the library staff have also become proficient in its use. The following detailed explanation of BDEN is taken from BALLOTS Documentation Standard DS.124.

* * * DOCUMENT FOLLOWS * * *

The BDEN system of notation has been developed to provide a standardized short-hand method for the precise description of data elements. BDEN notation takes the form of productions that look similar to equations. There are two classes of productions, one that defines the internal format of individual data elements, and a second that describes the use of data elements in on-line displays and printed outputs. The first class always contains a data element mnemonic as the left side of the production. The second class always contains a display or print position as the left side of the production.

ex: CLASS I
(IDX)::=<NUM(1-6)><NUM(1)CD>.<NUM(2)>

CLASS II
<4,3>::=(TUT)

I. A group of special symbols, < > ( ) ' ::=, is used to indicate the functions of various elements of BDEN descriptions.

1. < > ANGLE BRACKETS

Angle brackets around the left side of a production signify that the production is describing data in display format.

```
ex:  <IDX>::=
     <4,1>::=
```

Angle brackets in the right side of a production enclose the description of required data that is variable in content.

```
ex:  ::=<ALPHA(2)>
     ::=<TST>
```

2. ( ) PARENTHESES

Parentheses around the left side of a production signify that the production is describing data in input format.

```
ex:  (IDX)::=
     (4,1)::=
```

Parentheses within the left side of a production enclose information concerning the length of a field (see POSITION COORDINATES, II.13 below).

```
ex:  <4,1(10)>::=
```

Parentheses have two different uses in the right side of a production:

a. <()> When the parentheses occur within the angle brackets, the parentheses enclose information concerning the length of a unit of data (see LENGTH, II.6 below).

```
ex:  ::=<ALPHA(2)>
```

b. (<>) When the parentheses occur outside the angle brackets, the parentheses indicate that the data described within the parentheses is optional.

```
ex:  ::=(<NUM(3)>)
     ::=<NUM(3)>(-)<NUM(3)>
```

3. ' ' SINGLE QUOTES

Single quotes around the left side of a production signify that the production is describing data in print format.

```
ex:  'IDX'::=
     '4,1'::=
```

Single quotes in the right side of a production enclose one of the special symbols, < > ( ) | ' ::=, when these symbols represent literal data and therefore are not being used as special symbols within BDEN.

```
ex:  ::='('SEPN:<SP(2)><SEPN>')'
```

When single quotes are used as literals, they are written: '''

4. ::= DOUBLE COLON EQUALS

This notation separates the left and right sides of a BDEN production.

```
ex:  BAC::=<ALPHA(3)>-<NUM(3)>
```

5. | LOGICAL OR

Logical or's around the left side of a production signify that the production is describing data in storage format.

```
ex:  |IDX|::=
     |4,1|::=
```

The logical or on the right side of a production indicates exclusive alternatives in the form and content of data.

```
ex:  ::=<MEPN>
       |<MECA>
       |<MECF>
       |<MET>
       |<MEUT>

     ::=<ALPHA(3)>-<NUM(3)>
       |<ALPHA(3)>(<SP(1)>)<NUM(3)>
```

II. The various aspects of data elements are described according to the following conventions.

1. ADDITIONAL CHARACTER SET

(See CHARACTER SET, II.2 below.)

2. CHARACTER SET

(See also ADDENDA, below.)

The character set of the data being described may be indicated by using one of the following codes.

ALPHA	=	letters A-Z
ALPHANUM	=	letters A-Z and digits 0-9
BIN	=	digits 0-1, to be read as binary numbers (base 2)
CHAR	=	letters A-Z, digits 0-9, and all special characters
DEC	=	digits 0-9, to be read as decimal numbers (base 10)
HEX	=	digits 0-9 and A-F, to be read as hexadecimal numbers (base 16)
NUM	=	digits 0-9
SP	=	space
SPEC	=	all special characters i.e. all characters that are not letters or digits. This includes all punctuation and diacritical marks.

It is possible to specify a combination of two character sets. The ADDITIONAL CHARACTER SET must be separated from the initial CHARACTER SET by a comma. The only two codes that may be used for ADDITIONAL CHARACTER SET are "SP" and "SPEC." If, for example, a field could contain only alpha characters and spaces, it could be described as:

ex: ::=<ALPHA,SP(4)>

When character sets are linked together, they must be separated by commas.

If none of the above codes is sufficient, an EXPLICIT CHARACTER SET can be listed. If, for example, a field could contain either a "Y" for "YES" or an "N" for "no", and nothing else, the character set could be described:

ex: ::=<YN (1)>

The EXPLICIT CHARACTER SET must be separated from the designation of LENGTH by a blank. If a field could contain only alpha characters, periods, and commas, question marks, and parentheses, it could be described as:

ex: ::=<ALPHA, .;,?() (4)>

In the above example, the first comma separates the CHARACTER SET code "ALPHA" from the EXPLICIT CHARACTER SET. There should be no comma between the individual members of EXPLICIT CHARACTER SET. Also, there should be no comma between the EXPLICIT CHARACTER SET and LENGTH, but there must be a blank. If the EXPLICIT CHARACTER SET is very large, it is better to list it in a table than to describe it all within BDEN. The NAME convention can be used to facilitate this (see NAME, II.11 and III.1 below). Any indication of character set is always followed by LENGTH, enclosed in parentheses (see LENGTH, II.6 below).

3. COLUMN NUMBER

(See POSITION COORDINATES, II.13 below.)

4. EXPLICIT CHARACTER SET

(See CHARACTER SET, II.2 above.)

5. FORMAT

BDEN allows for the definition of four formats for a data element. These are input format, storage format, display format, and print format. The format being described by any BDEN production is indicated by the special symbols around the left side of the production.

ex: (XXX)::= describes an input format

|XXX|::= describes a storage format

<XXX>::= describes a display format

'XXX'::= describes a print format

6. LENGTH (OF DATA)

The length of a certain unit of data is indicated by placing a number in parentheses after the indication of the CHARACTER SET. The number of characters must always be preceded by an indication of CHARACTER SET. The number may be a fixed length or a range.

ex: ::=<ALPHA(3)>
::=<ALPHA(1-50)>

Use (n) to indicate an undefined range.

ex: ::=<NUM(n)>
::=<NUM(3-n)>

If the range or number of characters is not presently known, but will be known at the time a processing rule is executed because of previous processing, use a star to indicate the number.

ex: ::=<NUM(1-*)>

Zero is not a valid content for LENGTH.
If a unit is optional, use parentheses to indicate that fact (see I.2.b above). When writing numbers, do not use commas to separate hundreds from thousands.

ex: 1000 (not 1,000)

7. LENGTH (OF FIELD)

(See POSITION COORDINATES, II.13 below.)

8. LINE NUMBER

(See POSITION COORDINATES, II.13 below.)

9. LITERAL VALUES

A literal value is data that is fixed in content and in form. A literal value is noted without any conventional symbols surrounding it unless one of the conventional symbols is a part of the literal value. In the latter case, enclose the conventional symbols with single quotes.

ex: ::=DOG
::='('CAT')'

When single quotes are used as literals, they are written: '''

10. MNEMONICS

When a mnemonic is to represent the value of the data that the mnemonic names, enclose the mnemonic with angle brackets.

ex: ::=<IDX>

To describe the literal value of a mnemonic, treat it as a literal (see LITERAL VALUES, II.9 above).

ex: ::=IDX:<SP(2)><IDX>

11. NAME

Any unit of data enclosed by angle brackets on the right side of a CLASS I production (see BDEN PRODUCTIONS, III.1 below) can be named for ease of reference and identification. Use some simple abbreviation and place it after the designation of LENGTH. A name may not contain blanks.

ex: ::=<NUM(1)CD>

Explanations of the functions of various named units and the interrelation of these units may be given in notes and processing rules.

12. NUMBER OF CHARACTERS

(See LENGTH, II.6 above.)

13. POSITION COORDINATES

(See also ADDENDA, below.)

Position coordinates describe fields on screens, print formats, and other I/O documents. Position coordinates do not describe the structure, interior format, or length of data elements that occupy the fields in these I/O formats. (Compare this paragraph with LENGTH, II.6 above.) Position coordinates are described on the left side of a Class II production (see BDEN productions, II.2 below). A position coordinate may have four elements: (1) line number; (2) column number; (3) field length; (4) signal to right-justify data in the field.

model: LINE,COLUMN(FIELD LENGTH)SIGNAL

ex: '4,3(10)RJ'::=

Line number is usually indicated by some integer. If only a relative position is known, the symbol "L" may be used to represent current line number. The statement "Start in column four of the next line" is written:

ex: L+1,4

Relative columns may be treated in the same manner, using "C" to represent current column number. The indication of column is always separated by a comma from the indication of line. An indication of line and column, which defines the beginning of a field, is the minimum allowable content for a position coordinate. To this may be added the length of field. Field length cannot be expressed unless line and column are specified. Field length is always enclosed in parentheses, and is always one fixed number, not a range. When field length is specified, it can be followed by the code "RJ," which means right justify data in this field. Otherwise, left justification is assumed.

14. SPACE

A space is described <SP(1)>.
Three spaces would be described <SP(3)>.

III. BDEN PRODUCTIONS

There are two classes of BDEN productions.

1. CLASS I deals with the internal structure of a data element value. It describes the various units of information in a data element value by specifying the character set and number of characters in each unit of data, and by indicating whether a unit is a required or an optional part of the data element.

CLASS I model:

Data element mnemonic::=
<char set, add char set, explicit char set (length) name>

The data element mnemonic on the left side of the production will always be enclosed by one of the sets of special symbols to indicate which format the production is defining (see FORMAT, II.5 above). On the right side of the production the angle brackets enclose a required unit of data of variable content. A unit must contain at least one specification of CHARACTER SET and a specification of LENGTH. There may be any number of these units in a data element description. Units of literal data are placed in their proper position between the variable units. The literal units are not enclosed in angle brackets. A name is not required for the variable data. Literal data cannot be named.

CLASS I example:

(IDX)::=<NUM(1-7)><NUM(1)CD>.<NUM(2)>

This description means that the input format for data element IDX consists of four units, all of which must be present. The first element is composed of one to seven digits. The second is one digit, named CD. The third element is a period. The fourth element is composed of two digits. The purpose in naming the second unit CD is to facilitate reference to that unit in processing rules. For IDX there must be a processing rule that explains that unit CD is a mod 11 check digit on all the numbers that precede it. If there are several alternate formats that are acceptable, they may be shown by using the logical or.

```
(BAC)::=<ALPHA(3)>-<NUM(3)>
       |<ALPHA(3)>(<SP(1)>)<NUM(3)>
```

2. CLASS II deals with the use of data elements in I/O formats. It describes various fields in an I/O format by specifying the position of the beginning of the field in the format and indicating the data to be associated with that field.

   CLASS II model:

   ```
   position::=data description
   ```

   Position is defined by using POSITION COORDINATES (II.13 above). These coordinates will always be enclosed by one of the sets of special symbols. These symbols indicate the format type of the data described on the right side of the production (see FORMAT, II.5 above). This indication of format applies only to the form of the data element. It does not apply to the form of the I/O document being described. If data element IDX were for some reason to be displayed in its input format on an output screen, the production would be written:

   ```
           (4,13)::=<IDX>
   not
           <4,13>::=<IDX>
   ```

   The angle brackets around the mnemonic in the above example signify that the field cannot be blank.
   If the data is optional, the mnemonic will be enclosed by parentheses.

   ```
   ex:     (12,1)::=(RSAD)
   ```

   If a field must contain some special characters in addition to the value of the data element, these characters are described as literal data.

   ```
   ex:     <3,4>::=PRO:<SP(2)><PRO>
           <19,1>::='('SEPN')'
   ```

In many cases, however, such special symbols may already be a part of the data element format.
If a field may contain any one of a group of data elements depending on which one is present in the record, the alternatives are listed using the logical or.

ex: <9,1>::=<MEPN>
|<MECA>
|<MECF>
|<MEUT>
|<MET>

ADDENDA

II.2 CHARACTER SET

For use in describing control characters on the Sanders terminal, add the following symbols to the list of character sets:

C/R = carriage return character
H/M = home character
H/T = horizontal tab character
V/T = vertical tab character

A single horizontal tab would be described as:

<H/T(1)>

II.13 POSITION COORDINATES

Screen coordinates must indicate the BLOCK, as well as the LINE and COLUMN number. BLOCK is indicated by a single alpha placed before the LINE number and separated from that number by a comma.

(A,1,1)

This notation describes the upper-left-most character position on the screen and indicates that this position is in block A. Line and column are read to mean absolute screen position, not relative block or buffer position. A location at line 6, column 3, in block C would be described:

(C,6,3)

even through this might be the first position in block C.

* * * DOCUMENT ENDS * * *

3.6 PROGRAMMING LANGUAGE CHOSEN FOR BALLOTS II

Portions of BALLOTS I were coded in PL/1 and portions were coded in Assembly language for the 360/67 computer. It was felt that there were serious drawbacks to each of these languages and an investigation of available languages was conducted. PL/1 had the advantage of generating code quickly, but unfortunately the core requirements and execution time for the code were a substantial disadvantage. Assembly language, on the other hand, allowed efficient utilization of core and machine cycles but took an extended length of time to program. After an extensive survey of available assemblers and compilers, PL360 was chosen. PL360 is an algol-like language that is still close enough to assembly language coding to allow the user to control the use of computer registers. The PL360 language has essentially all the power of assembly language and yet allows the programmer to concentrate on the essential programming tasks. The following SPIRES document gives the "Coding and Description Standards for the Use of PL360."

* * * DOCUMENT FOLLOWS * * *

1. Introduction

This is a description of the placement of code and comments on the page listing; the nature of auxiliary descriptive text; preferred language constructs for efficient, compact object code; and control of source code versions.

2. Hierarchy of Indentation

PL360 is a block-structured, procedure-oriented language: logically related sequences of statements may be grouped and nested by means of the delimiters BEGIN and END. The COMMENT symbol allows descriptive remarks to be inserted anywhere within the code (except in literals) where a blank may be placed; the COMMENT symbol and following characters through the next semicolon are ignored by the compiler.

Although PL360 is free-field, this property should be exploited to define a semi-fixed format that helps the eye, with its pattern recognition ability, to grasp the organization of the program. (The same goal is further advanced by using FOR, CASE, and WHILE statements wherever possible to replace the IF-GOTO combination.)

The basis for the format is the INDENTATION LEVEL. For concreteness, the left-hand margin is taken as level 0, and each higher level is five columns to the right of the previous level. A CONTINUATION COLUMN is two columns to the right of the current indentation level. Since literals must be continued from the left-hand margin they should, if possible, be kept short enough to fit on one line by splitting them or by starting a new line before beginning them.

In the following paragraphs, DUMMY BASE can be substituted for BEGIN, and CLOSE BASE can be substituted for END.

BEGIN and END are always the first symbol in any line in which they appear (exception: labels, discussed below). BEGIN signals the last line of the ith indentation level with succeeding lines to be governed by the (i+1)st level. Only BEGIN can increment the current indentation level. END signals the first line of the ith indentation level following zero or more lines at the (i+1)st level. Only END can decrement the current indentation level. Thus, matching BEGIN-END pairs can be immediately identified. In general, at most one complete statement or declaration appears on a line; however, several short, logically related statements may appear on a single line. If it is necessary to continue a statement on following lines, then they start in the continuation column of the current level; this rate is superseded by the begin-end criterion. Thus:

```
FOR R2:=R1 STEP 4 UNTIL R3 DO
  COST(R2):=QUANT(R2)*PRICE(R2);
  COMMENT: CALCULATE CHARGES FOR M CUSTOMERS;
```

but

```
FOR R2:=R1 STEP 4 UNTIL R3 DO
BEGIN R6:=X2(R2)+X3(R2);
     R6:=R6 SHLA 2;
END; COMMENT: NOW R6 CONTAINS THE X FIELDS IN
  PACKED FORMAT;
```

similarly,

```
CASE R4 OF
BEGIN COMMENT: PERFORM OPERATION 1, 2, OR 3;
     BEGIN COMMENT: OPERATION 1;
          <stmt>; <stmt>; <comment>;
          <stmt>; <comment>;
          <stmt>; <stmt>; <comment>;
     END;
     <stmt>; COMMENT: OPERATION 2;
     BEGIN COMMENT: OPERATION 3;
          <stmt>; <stmt>; <comment>;
          <stmt>; <stmt>; <stmt>; <comment>;
     END;
END;
```

A semicolon is not required after BEGIN. END should be followed by a semicolon except when it is immediately followed by ELSE.

internal procedure declarations are governed by the current indentation level. Labels, which must be easily locatable in the code, are handled as follows: regardless of the current indentation level, a label definition always starts at the left-hand margin. For example

```
THERE: <stmt>; <comment>;
     <stmt>; <comment>;
     BEGIN <decl>; <comment>;
HERE:     <comment>;
          <stmt>; <comment>;
          BEGIN <stmt>; <comment>;
               IF = THEN GOTO HERE;
               <stmt>; <comment>;
          END; GOTO DONE;
NEXT:     <stmt>; <comment>;
          <stmt>; <stmt>; <comment>;
DONE:
     END; <comment>;
END;
```

A long label definition may overlap into the current indentation level, forcing the beginning of the statement to the right. The colon should be followed by at least one blank. If a BEGIN or END would be forced out of position by a label definition, then the definition is placed on a line by itself prior to the BEGIN or END.

3. Comments

Comments embedded in the source listing are one of the most important parts of program documentation for debugging, system integration, and maintenance. A well-documented program may be as much as 40 to 50 percent comments.

Comments have several functions. A comment may: describe the purpose and general structure of a procedure; identify the use to which a particular declaration will be put; describe the (supposed) state of the program (variable values, register contents) at a certain line of code; specify the intent of one or several lines of code; convert an executable statement used for debugging into a self-history; or justify a design decision about a construct or algorithm used, and why alternative choices were rejected.

The position of a comment in the listing gives a clue to its scope. A comment centered on the page or consisting of asterisks across an entire line separates segments, global procedures, or major sections of code. A comment starting at the current indentation level or immediately after BEGIN applies to code following it. One starting after a statement on the same line, or in the continuation column of the following line, applies to that statement or to the state of the program after that statement. A comment immediately following THEN or DO describes the consequences of the satisfaction of the condition prior to the THEN or DO; a comment after ELSE describes the consequences of the failure of the condition:

```
IF ¬= THEN COMMENT: SEARCH FOR ANOTHER MATCH;
BEGIN COMMENT: INCREMENT I AND J;
     <stmt>; <comment>;
     <stmt>; <comment>;
END ELSE COMMENT: STORE ADDRESS OF MATCH;
BEGIN <stmt>; <comment>;
     <stmt>; <comment>;
END; COMMENT: SEARCH DONE. @T IS IN ALPHA;
```

4. Declarations

Declarations are systematically ordered and grouped within a global procedure to lessen the possibility of error and to make them easy to locate. The order should be as follows:

1. Register name synonym declarations, ordered either alphabetically or by register number.
2. External procedure declarations, ordered alphabetically.
3. Dummy base declarations.
4. Function declarations.
5. Long real cell declarations.
6. Real cell declarations.
7. Logical cell declarations.
8. Integer cell declarations.
9. Short integer cell declarations.
10. Byte and character cell declarations.

Cell declarations that overlap each other, or must be grouped in adjacent locations, may be ordered differently, but this fact must be made clear through comments.

Programs with large numbers of register and cell identifiers make use of the following naming convention: the first letter of each identifier names the type, and the remainder is arbitrary, generally a meaningful mnemonic. Exceptions are predeclared and external interface identifiers such as dummy base variables, and functions or procedures.

- F long and short floating-point registers.
- R general-purpose registers.
- E real cells.
- D long real cells.
- L logical cells.
- B,C byte or character cells.

```
H  short integer cells.
I,J,K,M,N  integer cells.
```

5. Efficient Constructs

To clear a register:

```
F0:=F0-F0;
F01:=F01-F01;
R6:=R6-R6;
R6:=ZERO; (When ZERO is R0 and contains 0.)
```

Use the following functions:

```
FUNCTION REDUCE(6,#0600),
  SETUP(6,#0500),
  BRANCH(15,#47F0),
  ADDR(2,#4100),
  SETZONE(8,#96F0);
```

REDUCE is a BCTR Rx,0 instruction. For example, REDUCE(R1) is preferred over R1:=R1-1;

SETUP is a BALR Rx,0 instruction. Thus SETUP(R1) sets R1 to the address of the halfword following the current instruction, usually another instruction.

BRANCH is a BC 15,<indexable address> instruction. BRANCH(B2(R1+4)) causes a jump to the location given by R1+R2+4. Usually SETUP is used to establish one of the registers and the other is an index.

ADDR is an LA Rx,=<literal value> instruction. For example, ADDR(R1,"-HI THERE SAM.") sets R1 to the address of the character string "-HI THERE SAM."

SETZONE is an OI instruction that OR's in a hexadecimal F in the zone portion of the addressed byte, e.g., SETZONE(B2(3)).

Specify an n-way branch on R2, where R2 may contain 0 through n-1:

```
R2:=R2 SHLL 2;
SETUP(R1);
BRANCH(B1(R2+4));
COMMENT: BRANCH CHOICES START HERE;
GOTO A; COMMENT: R2=0;
GOTO B; COMMENT: R2=1;
GOTO C; COMMENT: R2=2;
...
GOTO X; COMMENT: R2:=N-1;
COMMENT: END OF BRANCH CHOICES;
```

Set R1 to the address of an error message specified by R2, which has a value 0 through n:

```
IF R2>N OR R2<0 THEN R2:=ZERO;
COMMENT: NEXT N+3 LINES MUST NOT BE MOVED;
R2:=R2 SHLL 2; SETUP(R1);
EX(R0,B1(R2+8)); GOTO X;
ADDR(R1,"MESSAGE 0"); COMMENT: R2=0;
ADDR(R1,"MESSAGE 1"); COMMENT: R2=1;
...
ADDR(R1,"MESSAGE N"); COMMENT: R2=N;
X:  COMMENT: R1 CONTAINS ADDRESS OF SELECTED MESSAGE;
```

Convert value of R1 to decimal, limited to 4 digits, and put it at the address specified by R2:

```
COMMENT: WORKAREA MUST BE 8 BYTES ON DOUBLE WORD
  BOUNDARY;
CVD(R1,WORKAREA); UNPK(3,7,B2,WORKAREA);
SETZONE(B2(3));
```

Move memory FORWARD with possible overlap, where Rx is the number of bytes to move, Ry is the source address, and Rz is the destination address:

```
WHILE Rx>0 DO
BEGIN REDUCE(Rx);
     IC(Ra,By(Rx));
     STC(Ra,Bz(Rx));
END;
COMMENT: Ra IS AN INTERMEDIARY FOR THE BYTES;
```

Some short programming tips:

Use Rx:=Rx++Rx instead of Rx:=Rx SHLL 1.

Use Rx:=@Bx(constant) instead of Rx:=Rx+constant whenever the constant is in the range 0 through 4095.

Use Rx:=@Bx(Ry+constant) instead of Rx:=Rx+Ry+constant, provided the constant is in the range 0-4095.

Use Rx:=NEG constant instead of Rx:=-constant.

To blank up to 256 bytes use

```
MVI(" ",cell); MVC(n,cell(1),cell);
```

where n is 2 less than the total number of bytes to be blanked.

Caution: MVC, CLC, UNPK, PACK, ED, TR, TRT, OC, NC, and other instructions that have length fields require the length to be

1 less than the actual length desired. Thus MVC(0,B,A) moves 1 byte from A to B.

6. Source Code Control

This section discusses the separate listing to be maintained for each global procedure, and the rules governing modification of these listings.

The SHELL consists of the minimum declarations, delimiters, and code necessary to form a dummy module that can be compiled and link-edited into the system but that performs no function except to return. Once the shell has been approved there should be no need to make any changes to it, except in the event of changes in the design of system interfaces, including register conventions, parameter passing, etc. The shell is fully commented and becomes part of the module documentation.

The PRODUCTION VERSION is the current approved module actually in use in the system. Very rigid controls must be exercised over the modification of this source code. It contains comment-form debugging statements from the test and integration phases. If any approved changes have been made since the last complete rewrite, the old versions of affected statements are present in comment form.

The TEST VERSION is the programmer's working version, which he is free to experiment with and modify at will. Most frequently it will differ from the production version only with respect to trial bug corrections or other improvements.

7. Auxiliary Descriptive Text

Each module has printed maintenance information intended as an extension of comments in the source listing. This AUXILIARY DESCRIPTIVE TEXT is not intended as a design specification or functional specification. Where necessary, references to these other documents can be made. Material appropriate to this document includes, but is not limited to: overall philosophy and layout of the program; mathematical analysis relating to the specific algorithms; running-time analysis, both expected and worst-case; and core usage. Frequent references to code line numbers IN THE PRODUCTION VERSION should be made where appropriate.

PL360 is supported on the Campus Facility computer and has been documented by SHARE, the IBM User Organization. Since the PL360 language was chosen for use on the BALLOTS Project, the compiler has been modified to run in an interactive mode allowing the programming staff to code and execute program at one sitting.

* * * DOCUMENT ENDS * * *

3.7 NETWORK FEASIBILITY STUDY

The results of a preliminary cost study for the BALLOTS system (see section 3.8) showed that a very high percentage of the operating cost of the BALLOTS system would be in the overhead area; that is, the area required for operational continuity of the system (updating of files, storage of files, etc.), independent of the amount of usage of the system. An additional study was made to determine what the costs would be if the amount of BALLOTS transactions was doubled (i.e., an increase of 100 percent of all the file searching, printing of catalog cards, etc.). This study showed that the operational cost for doubling the transaction load would increase the operating cost by approximately only 59 percent. It therefore seemed feasible to encourage other libraries to use the system and take advantage of the incremental cost of the additional work load.

It was decided to investigate the possibility of a library automation network for local Bay area colleges and universities. Four local colleges and universities participated in a 13-week workshop to determine the benefits, costs, impact, advantages, and disadvantages, of using BALLOTS in their libraries. The workshop was from April into July, 1971. The results of this workshop were presented in a two-volume, 250-page report that went to the library directors. The report was considered confidential, in that it contained individual library costs, so this report was not generally distributed. A summary report that included details of the cost methodology and some of the results of the workshop was written in a forms that could be distributed. This summary report is contained in Appendix D. Following is the text of a paper delivered at the 1971 conference of the American Society of Information Sciences. It contains a brief summary of the activities covered in the workshop.

* * * DOCUMENT FOLLOWS * * *

AN ON-LINE NETWORK--COOPERATIVE PLANNING WITH SEVERAL LIBRARIES

A. H. Epstein, Douglas Ferguson, and Eleanor Montague
Stanford University
Stanford, California

Abstract

A cooperative feasibility workshop, involving five colleges and universities, was conducted at Stanford University over a period of 12 weeks. Its purpose was to determine the costs and

benefits involved in the use by a network of a library automation and information retrieval system being developed at Stanford. After an orientation that promoted open and continuing communication, the participants worked together to produce the tools and techniques for a feasibility analysis. This analysis established the network operating environment, the impact of participation and system benefits on user libraries, and the operational costs, including computing, manual, and displaced costs. The results of the workshop were presented in the form of a feasibility report to the library administration of each institution to aid management decision making. The methods that the study team developed, the work assignments, some informal findings, and observations on the workshop are presented here.

Introduction

Stanford University is developing a large, on-line, interactive library automation (BALLOTS)* and information retrieval system (SPIRES)** to operate in a production environment. CRT terminals will be used for library on-line input and display, and typewriter or CRT terminals will be used for information retrieval. Three main types of on-line files are being designed for library use: (a) MARC--Machine Readable Catalog--bibliographic data from the Library of Congress; (b) IPF--In Process File--records of the status of each book ordered by the library but still in the processing cycle (not available for patrons' use); (c) Catalog and Circulation files--records of the location and status of each book in the collection.

The system is being implemented in a series of modules. Each module permits additional types of book material to be processed or adds new on-line files. The first module being implemented (BALLOTS-MARC) includes an on-line MARC file of the most recent 6 to 12 months of MARC data for use in the technical processing of all books covered by the MARC data. The printed output of this module includes purchase orders and catalog card sets ready for filing.

The system is designed so that it can be extended to specialized libraries at Stanford, such as the law library, as

----

* Bibliographic Automation of Large Library Operations using a Time-Sharing System (1). BALLOTS was partially funded by grants OEG-1-071145-4428 and OEG-0-70-2262 from the U.S. Office of Education.

** Stanford Public Information Retrieval System (1). SPIRES is funded by a grant from the National Science Foundation.

----

well as to a regional network of libraries. From the point of view of a potential user library, joining the network has the following advantages over individual automation: (a) most of the development and installation cost and time are eliminated; (b) the operational cost is reduced by time sharing on a large computer; (c) stable software and hardware, which will have run under heavy production conditions at Stanford for four to eight months before network processing starts. Other advantages are shared by both Stanford and the user libraries: (a) shared BALLOTS overhead computing costs; (b) the opportunity of access to all files; as a result of this, the possibility of (c) sharing a portion of the technical-processing workload; (d) reducing future redundant acquisitions; and (e) increasing future interlibrary loans.

The Feasibility Study

In order to inform San Francisco Bay area schools about BALLOTS and to explore the possibility of a regional library automation network, Stanford invited library directors from several nearby colleges and universities to meetings in July 1970 and February 1971.

The participants in these meetings explored the possible usefulness of BALLOTS to other libraries. They agreed that a preliminary feasibility study was a prerequisite to a management decision to participate in BALLOTS. At the February meeting, four schools agreed to commit the necessary personnel and time to this study. Two senior librarians with technical processing experience were selected by each school. They met together with members of the BALLOTS staff at Stanford one day a week for 13 weeks from April to July. The equivalent of a second day each week was spent by team members at their own schools gathering data for a cost analysis.

Study Objectives

1. To understand how BALLOTS and, specifically, the BALLOTS-MARC system will support library technical processing.

2. To develop and test cooperatively a cost measurement methodology applicable within the time limits of the study. This methodology had to be uniformly applicable without placing a heavy data-gathering burden on the staff of each library. It had to permit comparison of the costs of existing manual procedures and the costs involved in the use of BALLOTS-MARC. The standardized methodology had to permit comparison between

libraries of the costs and times taken to produce a given unit or to perform a like activity.

3. To determine ways in which BALLOTS-MARC could be made of greater use to libraries outside of Stanford.

4. To prepare a report that would provide cost and benefit information for the Director of each library. This report would enable each library to determine the main implications and results of its participation in a BALLOTS regional network and would be sufficient for management to make a decision on whether or not to participate in a BALLOTS Network.

Study Methodology

The study team met as a group at Stanford for six hours every Friday for 13 weeks. At the first meeting, team members described their backgrounds and interests, the features of BALLOTS were reviewed, and each library was generally described by someone from that library. During subsequent sessions a cost measurement methodology was developed and applied. Each session ended with the assignment of tasks to be completed by the following session. At the next session, task results were reported and problems were discussed. Throughout, the emphasis was on creating a common framework for analyzing costs.

The feasibility study was not begun with a predetermined method for doing the cost analysis. A review of the literature on the subject indicated that a variety of methods has been developed under widely varying circumstances of library size, study objectives, validity requirements, and documentation to be produced. It was the purpose of this study to achieve some fairly specific objectives in a period of 13 weeks. An equally important intention was to encourage the study team to involve itself fully in defining the study method and to benefit from the critical comments and experience of all team members. The resulting costing methodology was the product of a joint effort by the entire feasibility study team.

Manual Costs

For manual costs, the approach taken was (a) to divide technical processing into 12 major functions; (b) to identify the component activities of each function for which costs could be calculated; (c) to design and test forms for collecting and calculating manual costs; and (d) to gather time and cost data.

Both total function cost and unit output cost were calculated for each major function.

A function was defined as a series of actions that produce an identifiable and measurable result (e.g., a purchase order or a cataloged book). The individual output of a function is the item to which a unit cost of production can be assigned. The 12 functions identified were: Ordering, Purchase Order Receipt, Non-Purchase-Order Receipt, Distribution (of material within technical processing), Obtaining and Maintaining Library of Congress (LC) Data, LC Cataloging, Original Cataloging, Card Production, End Processing, Claiming, Cancellation, and Added Copies and Volumes.

The component actions of a function are called activities. Activities are not always easily distinguished, and in different libraries functions that produce the same output may consist of different activities.

After the functions were identified, they were flowcharted in a team session in order to divide each function into its component activities. These charts were very simple block diagrams that labeled and sequenced the activities in a function. They reflected the slight differences in processing flow that existed among the participating libraries.

The major cost of an activity was identified as the cost of the people performing the activity. Personnel times for activities were collected for three categories of personnel: professional, assistant, and student. For each category of personnel at each library, an adjusted hourly wage rate was calculated based on productive hours (accounting for vacation, sickness, and breaks) and including average benefits for that library. As such, the adjusted rate is higher than the standard hourly rate. Supervisory costs were included in personnel costs only if the supervisor actually contributed time to the performance of the activity.

Costs for equipment (e.g. xerox toner) and supplies (e.g. card stock) were added to the personnel costs for each activity to yield total function costs. Costs for overhead, space, and general office supplies were not included. The purchase price and maintenance of equipment were not included. The unit cost for each function was obtained by dividing the total function cost by the number of outputs produced from that function.

The study team did not attempt in its methodology to balance total function costs within a department against departmental budgeted dollars. This could be done as a follow-up study and would have to take into account pockets of activity not accounted for in this study.

Manual-Automated Costs

There are four kinds of costs involved in the use of the BALLOTS network system. They are essentially costs for which the user will receive a bill each month from equipment manufacturers, from the telephone company, and from the Stanford Computation Center. The four costs are: on-line transaction costs for activities such as searching for or modifying a record; batch transaction costs for printing purchase orders, catalog cards, etc.; dedicated equipment costs for terminal rental and modem and telephone line charges; and overhead costs for file storage and updating and line connection charges.

To calculate a library's annual expenditure, it was necessary to know the library's volume of transactions against the system. To secure this data, a statistics questionaire was prepared that asked for specific figures on the amounts of data that would be processed through the BALLOTS-MARC system and on the number of outputs that would be produced. In conjunction with this questionnaire, a schematic diagram of the system was prepared for the team's use. The statistics questions were keyed to processing points on the diagram. All the libraries were able to supply transaction volumes based on actual counts or estimates.

Estimated annual dollar costs were calculated for each library by multiplying unit cost data for each type of on-line or batch transaction (supplied by Stanford) by the transaction volumes from each library. For each on-line or batch transaction, a maximum (conservative) and a minimum (average) unit cost were provided. The final operating unit costs of the system will fall somewhere in this range. Final calculations were made using both maximum and minimum unit costs. However, the summaries for each library were made using the maximum unit cost, in order to provide a conservative or worst-case cost estimate for management decision making. In BALLOTS, the two typical on-line transactions are searching to locate a record and modifying a record by adding or changing data to produce a printed output. To reflect variation, unit costs were calculated for simple, medium, and complex searches. Batch transactions produce printed library outputs and handle standing search requests against the MARC file.

Equipment and computer overhead charges were not allocated on a per unit basis. Equipment costs are the charges for dedicated equipment used exclusively by an individual user library. These include the visual display terminal, a modem, and leased telephone lines. Computer overhead costs are the sizable fixed costs that include the processing of major file updates, charges connected with file storage, and hardware connection time for the use of a small computer to handle communications between the terminals and the Stanford Computation Center 360/67. For the purpose of this study, computer overhead was allocated proportionally to expected system usage (not the same for each library) and on the assumption that the four feasibility study participants plus Stanford would make up the network. Of course, if more libraries join the network, each library's portion of overhead will decline; if fewer libraries join the network, each library's portion of overhead will increase.

The next step in the study was to recalculate costs for all the functions affected by BALLOTS-MARC. After discussion, eight of the 12 functions were selected as the ones most likely to be affected by BALLOTS-MARC services. The four functions not affected were Claiming, Cancellation, Added Copies and Volumes, and Original Cataloging.

Each of these eight functions was again diagrammed--as it would be performed under manual-automated conditions. Standard time rates were used by all libraries for the new manual-automated activities (e.g., input at a CRT or searching an on-line file). Equipment and computer overhead charges were not allocated to individual functions but added in at the library summary level. However, on-line and batch transaction charges were charged to the function for which they occurred.

Thus, the costs for a function involving BALLOTS-MARC were a combination of revised personnel and equipment costs for specific activities and of computing costs (on-line and batch charges). The total of these new function costs, plus terminal equipment and computer overhead charges, was the cost to the library of using the BALLOTS-MARC system. The personnel costs were usually reduced because computer processing replaces some manual processing. Computer costs, however, were an added cost factor.

The charges for on-line transactions, batch transactions, terminal and associated equipment rental, and computing overhead make up the amount paid in dollars each month by each library participating in the network. In general, this is an added cost to the library and represents added cash flow. But the difference between personnel costs under the current manual system and personnel costs under the manual-automated system is a function cost reduction that is, in effect, a saving to the library. Personnel savings do not affect the cash flow unless there is a net reduction in staff. Reassigning staff to new work is a benefit to the library in terms of the added service performed, but does not help to reduce the cash flow. Reassignment must be evaluated separately by each library, and may have an impact on the decision to participate in a network.

The Feasibility Report

At the conclusion of the study, a 250-page, two-volume report was prepared and distributed to the directors of the four libraries. The report covered all four libraries. Volume 1 contained two sections: general textual discussion of the purpose of the study, the BALLOTS-MARC system, and the library cost analysis methodology, and a discussion prepared by each library that summarized the findings of the cost analysis for that library, advantages and disadvantages. Volume 2 comprised appendices containing the actual cost forms, calculations, unit on-line and batch transaction charges, and overhead cost calculations. No library was specifically named in the report. The first section of Volume 1 was prepared by the BALLOTS staff and approved by all team members. The second section was prepared by the team members of each library for their library. It was the purpose of this second section to discuss the cost implications of using the BALLOTS-MARC system, the effects on the current system, and non-economic advantages and disadvantages. However, since no member of the team was in a library administrative position, this section did not attempt to offer suggestions for obtaining funding for the new cash flow. The feasibility report is summarized in one volume for the benefit of additional potential network libraries (2). The summary contains the methodology, unit costs, and advantages and disadvantages from Volume 1 and the sample forms and calculations from Volume 2.

The report did not attempt to draw overall cost and benefit conclusions from the studies done in these four libraries. The report, we think, supplies the director of each library with all the quantitative data needed to evaluate the use of BALLOTS-MARC: the dollar decrease in personnel costs, the cost of on-line and batch transactions (maximum and average), the CRT equipment and associated equipment rental charges, and computer overhead. Of these costs, only the amount of computer overhead charged to each library is affected by outside influences--namely, the number of libraries in the network. Everyone agrees that management must have cost data in order to decide whether to join a network or to undertake independent library automation. It is obvious, however, that cost is only one of many items to be considered. The final decision is based on a combination of factors, some of which are affected by the personalities of the people involved, by the library, by the college or institution to which the library belongs, and by the state of the economy. Other concerns are morale, expectations of future benefit, augmented services, and the desire to reduce current inefficiencies and problem areas. The second section of the report covered not only cost but also these other non-economic, intangible considerations.

This study reflects an approach to planning for a network that deserves further exploration. The potential members of a network (the user libraries) need to know if it is to their advantage to join the network. For the most part they do not know how to ask questions that will give them this information.

One approach is to hire a consultant. This may have the disadvantage of requiring a period for orienting the consultant to individual library operations and for training some portion of the staff in the consultant's "method." Also, a consultant can produce concern about change being introduced from without the library.

The BALLOTS approach was to form a team of working librarians from all the libraries. The team developed the methodology and applied it. Since the team members had several years' experience in their respective libraries, they could efficiently secure the needed information and explain to their colleagues the purposes of the study. The study was completed in a relatively short time with no adverse staff reactions. Another effect of using a team of working librarians was to involve the library staff in automation decision making at the earliest (i.e. feasibility) stage. For the administrator, the section of the report that his own staff produced, the data and the analysis, adds to the credibility of the report.

Next Steps

To proceed with an on-line library network, a second workshop will be conducted with interested libraries. To follow the existing cost analysis and feasibility report, the next task is to determine individual user library requirements. These requirements will specify in detail any additions or modifications required by each library to BALLOTS--MARC CRT screens, outputs, services, etc. The output of the second workshop will be a Library Network Requirements document. This document will also include the detailed plan, schedule, and cost of training, installation, and production operation for the network. Specific areas covered by the first workshop (such as installing the BALLOTS-MARC system in the present manual system) will be treated in greater detail, and new areas (such as forms design and acceptance testing) will be examined. For new libraries interested in joining the network, an abbreviated feasibility workshop is planned, since the cost methodology, unit costs, and sample calculations are available.

(1) System Scope for Library Automation and Generalized Information Storage and Retrieval at Stanford University, SPIRES/BALLOTS Project, Stanford University, Stanford, California, 1970.

(2) Montague, Eleanor, Summary of a Feasibility Study on the Participation of Four Colleges and Universities in a Stanford University Library Automation Network, SPIRES/BALLOTS Project, Stanford University, Stanford, California, 1971.

* * * DOCUMENT ENDS * * *

After the workshop was over and the reports had been reviewed by the four library directors, each library determined that the BALLOTS system showed sufficient promise that a second workshop should be organized to obtain more detailed information on necessary changes to the BALLOTS system to permit its implementation in their respective libraries. This second workshop is underway at the time of this writing, and to date few substantial changes to the BALLOTS system are foreseen as a result of the additional or modified requirements of the various libraries. One library has requested an additional data element to support an internal accounting system.

3.8 OPERATIONAL COST STUDIES

As soon as the BALLOTS preliminary modular design and the SPIRES/BALLOTS file services design (shared software) had been completed, calculations were made to determine the approximate operating cost of the BALLOTS-MARC module operating alone, as well as of the cost of the full BALLOTS system. The costs were broken down into three major areas: on-line transaction-generated costs; fixed monthly costs; and overhead costs to keep the system operational.

The on-line transaction-generated costs include the costs of the on-line transactions and the costs of the batch printing charges generated as a result of the on-line transactions. A unit cost was calculated for each different type of transaction or printing charge, and this was multiplied by the expected number of transactions. For example, the unit cost of a simple search was multiplied by the number of simple searches expected in one year. Figure 8 lists the various unit costs used by Stanford and the network libraries in their calculations. Figures 9 and 10 give the number of each type of transaction done by the library in one year's activity. The methodology used to estimate transaction loads is described in Appendix D. Figure 11 summarizes the results of Stanford's cost calculations for the BALLOTS-MARC module and for the full BALLOTS system. One method of measuring system costs (since documents are the primary product of the system) is to divide the entire system cost by the number of output documents to show the unit cost per document. Dividing the total BALLOTS system monthly cost of $27,155 by the number of output documents (70,200 per month) produces a cost of $0.39 per original output document.

The fixed monthly charges include the rental of the terminals and the leasing of the telephone lines and modems required for each terminal. In the case of the Stanford libraries, the leased cost for telephone lines is zero because Stanford has purchased and installed modems and telephone lines between the Library and the Computation Center. In the case of network libraries, modems are leased and the cost of the leased

(Note: each library is charged on the basis of its individual total monthly transactions.)

A. ON-LINE TRANSACTIONS

A high unit cost and a low unit cost are given; the actual measured results should fall somewhere between the two unit costs.

Transaction Type	High	Low
1. Simple search	$0.04	$0.02
2. Medium search	$0.17	$0.08
3. Complex search	$0.33	$0.17
4. Simple input	$0.10	$0.05

B. BATCH TRANSACTIONS

All printed outputs are figured at a high unit cost of $0.04 and a low unit cost of $0.02. In addition, each printed output involves a $0.02 print charge, making the effective high and low unit costs $0.06 and $0.04.

C. DEDICATED EQUIPMENT CHARGES (MONTHLY)

1. Terminal rental (stand-alone terminals)	lease $225. (ea.)
2. Terminal rental (clustered terminals)	lease $155. (ea.)
3. Data set rental* (one per terminal)	lease $ 50. (ea.)
4. Leased telephone lines** (one per terminal)	lease $ 3. (per mi.)

---

* Stanford will purchase data sets so the on going cost will be eliminated.

** Libraries on the Stanford campus do not pay telephone line charges since lines have been purchased.

Figure 8. Unit Costs of Computer Transactions for BALLOTS

1. Purchase Orders - 24,800

2. Abel Slips (Abel Original Invoice, Abel Accounts Receivable) - 15,000

3. NPAC (National Program for Acquisitions and Cataloging) Notices - 4,000

4. Process Slips (no accounting for multiple copies) - 5,000

5. Statistics Reports (Acquisition and Cataloging) - 12

6. Title II Slips - 4,250

7. Catalog Data Slips - 47,250

8. Requester Notices - 32,500

9. Claim Notices - 7,500

10. Cancellation Notices - 750

11. Claim Listings - 52

12. Cancellation Listings - 52

13. Invoice Claim Listings - 52

14. Invoice Claim Notices - 3,750

15. Claim Errors - 52

16. Requests for Books in Process - 1,500

17. Catalog Cards - 500,000

18. Spine Labels (2 per book) - 110,000

19. Book Cards - 55,000

20. Reference Cards - 13,000

21. MARC Purged Standing Search Requests - 2,740

22. MARC Standing Search Requests Matched Records - 14,520

23. MARC Data Slips - 250

Figure 9. Stanford Full Automated System Annual Printed Outputs

ON-LINE TRANSACTIONS (annual figures in thousands)

FUNCTION	S101* Search Input	OR01 Ordering Input	RX01 In Process Update	BI01 Biblio. Input	BI02 Biblio. Input	BU01 Biblio. Update	BU02 Biblio. Update	HHXX Holdings Input	HXXX Holdings Update	Est. Stand. Search Requests	RF01 Reference Input	TOTAL
1 - Ordering	61.0	24.5	24.5	9.0	4.0	1.0	.5					124.5
2 - Purchase Order Receipt	30.0	4.0	28.0	1.8	.2	1.2	.6			8.0		73.8
3 - Invoice Receipt	5.0		5.0									10.0
4 - Force Claim	2.0		2.0									4.0
5 - Non-purchase Order Receipt	42.0	17.0	17.0	13.0	5.0	1.0	.5			2.0		97.5
6 - CT	37.7		2.0	5.5	3.2	12.4	3.2	40.0	5.5	.2	3.1	117.[illegible]
7 - End Processing	45.0		45.0									90.0
8 - SSR Purge	1.5											1.5
9 - Maintenance	5.0					2.0	1.0	2.0				10.0
10 - Misc. Searching	50.0											50.0
TOTAL	279.2	45.5	123.5	29.3	12.4	17.6	10.8	42.0	5.5	10.2	3.1	579.1**

(Note: these calculations were made on the basis of the full system acquisition and cataloging function; but the total transaction activity by screen type holds good for the BALLOTS modules.)

* S101 search transactions breakdown:

	(per year)	(per day)
Simple	197.5	800
Medium	76.7	300
Complex	5.0	20
	279.2	1,200

**Total input and update transactions (excluding S101):

(per year)	(per day)
299.9	1,200

Figure 10. Stanford Full System Technical Processing Load (by Screen Type)

(Note: higher unit cost used in calculations)

TYPE OF TRANS-ACTION OR CHARGE	UNIT COST	BALLOTS-MARC MODULE COSTS		FULL BALLOTS SYSTEM COSTS	
A. <u>On-line transactions</u>:		no. of <u>trans.</u>	<u>cost</u>	no. of <u>trans.</u>	<u>cost</u>
1. Simple search	$.02/.04	35	$ 1.40	800	$ 32.
2. Medium search	.08/.17	90	15.30	300	51.
3. Complex search	.17/.33	42	14.00	20	6.
4. Simple input	.05/.10	174	17.40	1,200	120.
DAILY SUBTOTAL		341	$48.10	2,320	$209.
MONTHLY SUBTOTAL		7,500	$1,060.	51,000	$4,600.
B. <u>Batch transactions</u>:		no. of <u>docs.</u>	<u>cost</u>	no. of <u>docs.</u>	<u>cost</u>
1. P.O.'s	$.02/.04	54	$ 2.16	160	$ 6.
2. Cards	"	833	33.32	2,054	82.
3. Labels	"	249	9.96	44	18.
4. SSR's match & purge	"	236	9.44	69	3.
5. Title II slips	"	0		17	1.
6. Process slips	"	326	13.04	362	14.
7. Circ. punched cards	"	0		220	9.
8. Misc. forms	"	0		48	2.
CPU SUBTOTAL		1,698	$67.92	3,370	$135.
Print charges	$.02		$33.96		$ 67.
DAILY SUBTOTAL		1,698	$101.88	3,370	$202.
MONTHLY SUBTOTAL		37,356	$2,241.	70,200	$4,210

Figure 11. BALLOTS Cost Calculations (continued on next page)

(Note: higher unit cost used in calculations)

TYPE OF TRANS-ACTION OR CHARGE	UNIT COST	BALLOTS-MARC MODULE COSTS		FULL BALLOTS SYSTEM COSTS	
C. Dedicated equipment charges:		amount of item	cost	amount of item	cost
1. Terminal rental	($155/mo.)	3	$465.	15*	$2,330.
2. Data set rental		0	(purch.)	0	(purch.)
3. Dedicated line		0	(purch.)	0	(purch.)
MONTHLY SUBTOTAL			$465.		$2,330.
D. Overhead charges:		amount of item	cost	amount of item	cost
1. Batch util. & file update		--	$3,630.	--	$7,000.
2. File storage ($1,200/disk)		1	1,200.	7**	8,400.
3. Connect time ($3.50/hr.)		176	615.	176	615.
4. Circulation computer***		0	______	1	(purch.)
MONTHLY SUBTOTAL			$5,445.		$16,015.
TOTAL MONTHLY COMPUTING COST			$9,211.		$27,155.
TOTAL ANNUAL COMPUTING COST			$10,532.		$325,860.

* Clustered terminals.

** Storage costs: 6 disks the first year and 1 disk added per year (CDF); 1 disk added for Circulation.

***Assumes minicomputer purchased for Circulation transactions, but disks and terminals still leased (no additional on-line charges involved).

Figure 11. (continued) BALLOTS Cost Calculations

telephone line is a function of the distance from Stanford University.

The computer system overhead charges include the cost of file storage (i.e., storing the MARC file and the In Process File, etc.). The file update costs include the costs of reading and converting the weekly MARC tapes and adding this data to the on-line MARC file. Also included in these file update costs are the costs associated with updating individual records that are changed by the library during the course of the daily activity. The on-line connect cost is a fixed cost per hour for connecting the terminal to the Campus Facility computer. From Figure 11 it was determined that the costs due to on-line and batch transactions (A. plus B.) represent only 32 percent of the entire operational cost of the full system. The rough calculation shows that adding enough network libraries to the system to double the system throughput (the total of on-line and batch transactions) would double the cost of on-line and batch transactions (A. plus B.) and presumably double the cost of terminals (C.), while leaving the computer system overhead charges (D.) roughly the same. According to the data in Figure 11, the monthly cost for the Stanford full system excluding overhead charges is $4,600 (A.) plus $4,210 (B.) plus $2,330 (C.), totalling $11,140. Adding network libraries to double the amount of on-line and batch activity would mean an additional monthly cost of $4,600 (A.) plus $4,210 (B.) plus $5,180 (C.), totalling $13,990. The dedicated costs for network libraries include data set rental, line costs, and the unclustered (stand-alone) terminal costs. (The stand-alone terminal cost, $225 per month, was used in this calculation because many small libraries will find that clustered terminals do not suit their operations.) $11,140 per month for Stanford and $13,990 per month for network libraries plus $16,015 for the overhead charges (D.) plus $2,000 additional update and storage equals $43,145 per month. Dividing this cost by 140,400, the number of documents produced monthly (70,200 times 2), yields a cost of $0.31 per document. Comparing $0.31 with $0.39 and $43,145 with $27,155 indicates that a substantial reduction in unit costs can be achieved by increasing the system activity. While one is increasing the monthly operating costs by 59 percent (from $27,155 to $43,145), one is also doubling the system output and decreasing the unit cost of a document by 20 percent (from $0.39 to $0.31).

3.9 NET COST OF LIBRARY AUTOMATION

One of the major unknowns in cost calculations is the total number of libraries participating in the BALLOTS network. Since the overhead cost is the major factor in the cost calculations and in the ultimate costs to the various libraries, the more members participating in the BALLOTS network the lower the overhead proportion will be for each library and the easier it will be to justify library automation in individual libraries.

Most of the cost figures used in the report are of necessity calculations, not measurements of actual production operation costs. Until the system has been in operation for several months it will not be possible to arrive at accurate costs. In determining the net cost of library automation, many factors not yet mentioned must also be taken into account.

One of the considerations to keep in mind in the area of manual savings in the library is the fact that although it may be possible to save a substantial amount of manual activity, this savings in time cannot be translated to cash flow dollars by the library unless the positions affected are eliminated. Most libraries feel that attrition would be the only way to reduce the present staff, and this is a relatively slow process. Savings that appear as fractional personnel would probably result in reallocation of assignments to provide better service. Until attrition could eliminate a full position or more, to offset incremental costs of operating the automated system, the actual cash flow would not reflect any savings. It is for this reason that the calculations for cost savings due to library automation are gradually obtained over a period of several years rather than immediately on successful implementation of the system. One factor absent from these cost calculations was the increased cost of labor. The net increase in salaries for various libraries averages from 4 to 10 percent per year. If it were possible to add these factors into the cost calculation, the net cost of library automation would appear somewhat less expensive.

CHAPTER 4

ACTIVITIES FOLLOWING THE REPORTING PERIOD

4.1 IMPLEMENTING THE BALLOTS-MARC MODULE

Several activities are now underway or in the planning stages to prepare for the actual daily operation of the BALLOTS system--first at Stanford, then in the CLAN libraries. Almost all of these activities support the operation of the first module, BALLOTS-MARC.

Under the BALLOTS system, a new module may require creating a new on-line file or adding additional applications software to support an existing file. In either case, the implementation of a module requires integrating and coordinating six major systems analysis tasks and three major programming tasks. (The numbers following refer to Figure 12.) The systems analysis tasks are: (1) determining system requirements, (2) preparing representative test data, (3) preparing training and user manuals, (4) acceptance testing, (5) user training, and (6) pilot production. After (7) requirements review, the programming tasks are (8) programming (including design, coding, testing, and documenting), (9) system acceptance testing, and (10) pilot production.

These systems analysis and programming tasks are carried on in parallel in the following way to move the module towards implementation. (Again the numbers refer to Figure 12.) (1) The library analysts prepare system requirements jointly with the library and secure the library's approval. (7) These requirements are passed on to the programming staff for review as to their clarity, consistency, and design feasibility. While this review is going on, the analysts and programmers discuss the requirements and the programming to be done, and (2) the analysts begin preparing representative test data. After the requirements are given technical approval, (8) programming begins. After design, coding, and preliminary debugging, the test data are used with the new programs. While programming is being done, (3) the analysts plan user training and prepare training material. Using documentation prepared by the programmers, preliminary versions of user manuals are written. When programming is completed, (9) the programmers begin system testing while analysts (4) perform an acceptance test and (5) train user supervisors; then the library supervisors perform an acceptance test. User manuals are revised, (5) the full library staff is trained, and (6), (10) pilot production begins. System testing continues throughout pilot production and into the first few weeks of full production. Maintenance documentation is put in final form by the programming staff at the end of pilot production.

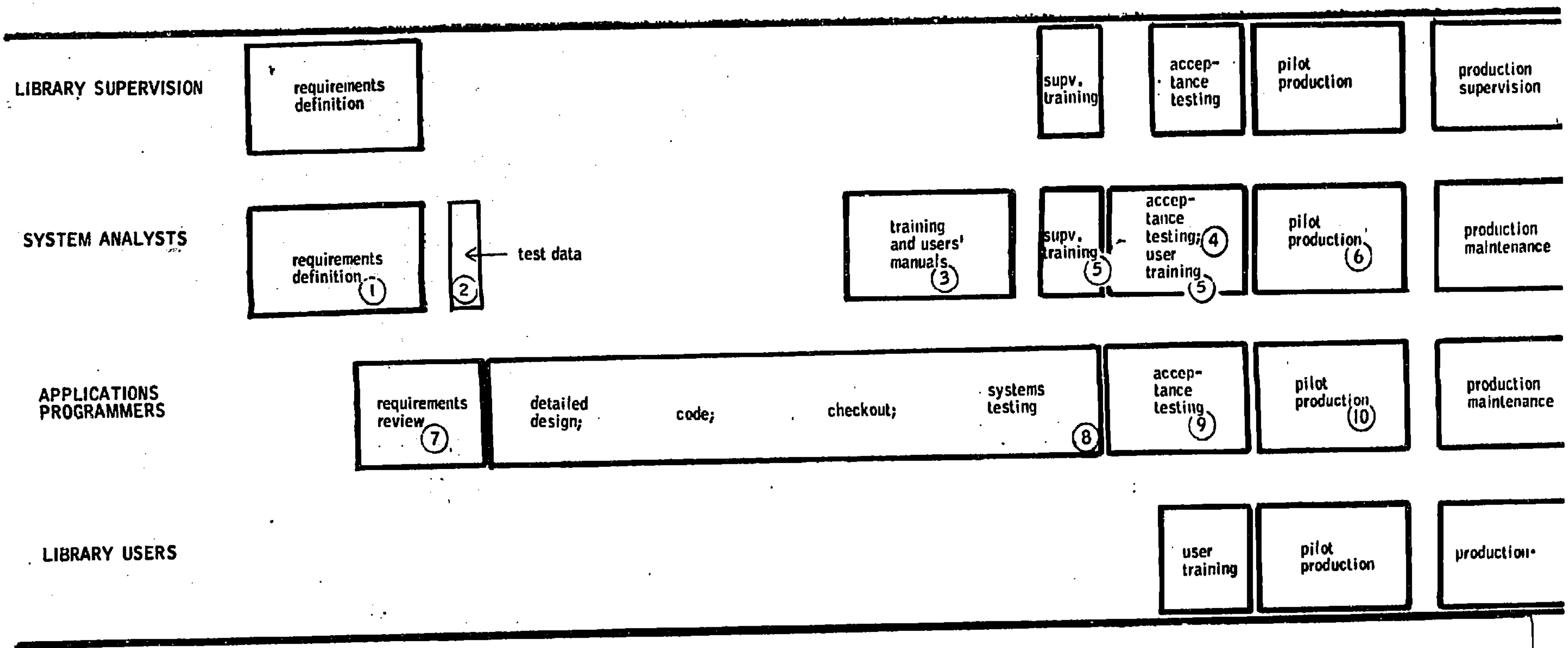


Figure 12. BALLOTS Module Implementation Tasks

A typical cycle of implementation tasks takes from 5 to 52 weeks depending on the module's complexity and the manpower assigned to the task. The first module and one of the largest is BALLOTS-MARC. BALLOTS library analysts and programmers are well into the BALLOTS-MARC implementation activity as this report is being produced. The ability to meet the projected schedule and to maximize the use of staff and hardware (and thereby to proceed most economically in a cost benefit sense) depends on external funds to support the completion and implementation of the BALLOTS modules during 1972 and 1973.

The library analysts are now writing instructional manuals for the users: the training and procedures manuals. The training manuals will give the basic background information including an overview of the system, as well as the technical knowledge required to operate the video terminals and use all of the various screen formats. The procedures manuals will show how the library's manual procedures are adjusted to allow the automation system to become part of the day-to-day environment of the library. The location of the video terminals in the Stanford Main Library has been determined and the communication lines are being installed. Once the terminals have been installed, the library analysts will check them out and begin user training in the Library.

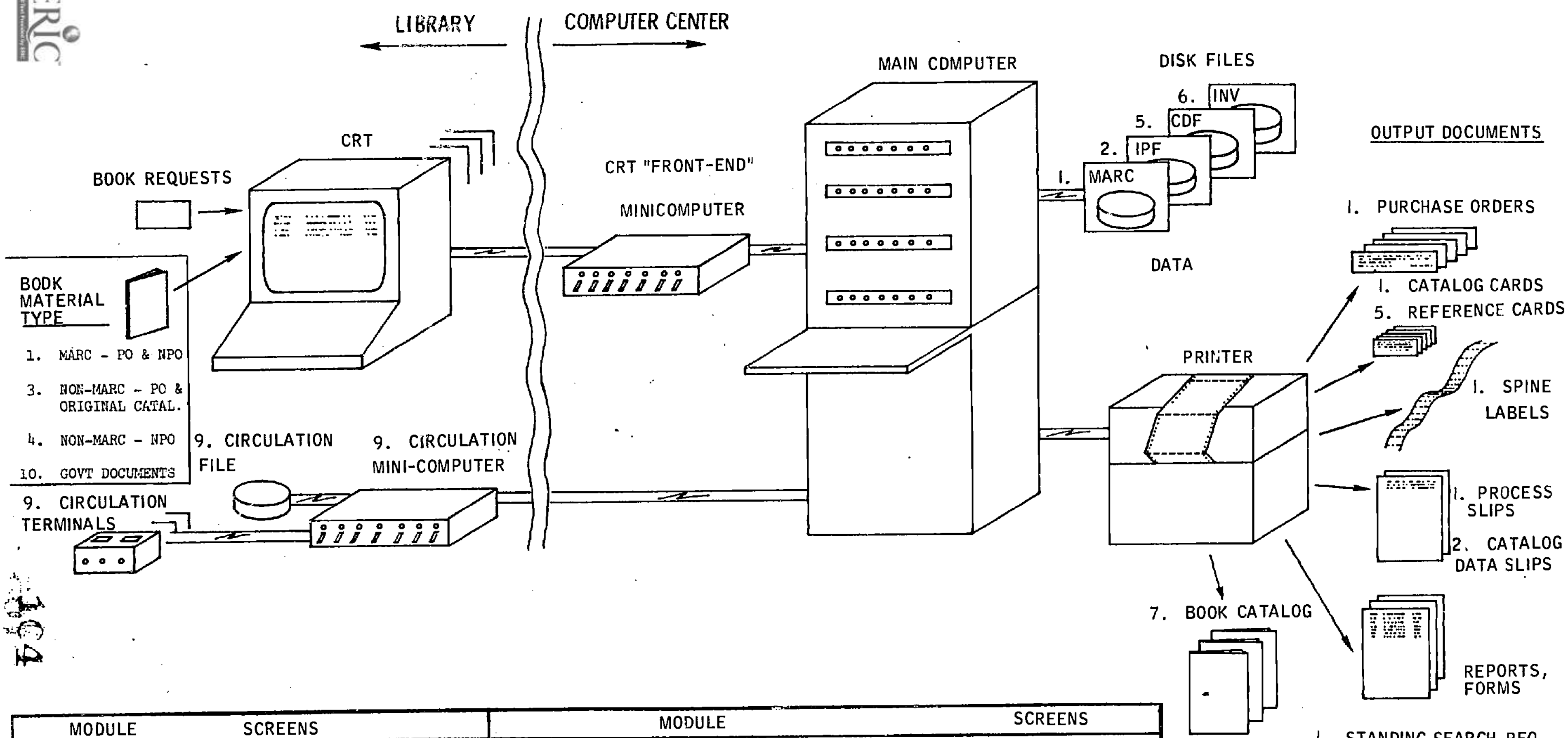
The BALLOTS programmers are completing the check-out of the BALLOTS subprocessor (see section 3.3) for the BALLOTS-MARC module. The CRT screen formats are being coded and checked out. In addition, the output documents required for the first module will be produced and checked.

Following this analysis and programming work, the acceptance testing, user training, pilot operation, and full operation will be conducted.

As described in the concluding paragraph of section 3.1.3, the Stanford Computation Center Campus Facility made a commitment to BALLOTS' implementation via a "front-end" minicomputer into MILTEN, the 360/67 communications subsystem, and ORVYL, the 360/67 time-sharing subsystem. This commitment meant ordering and installing additional hardware at the Campus Facility and making software changes to accommodate BALLOTS. Both activities are being carried out.

4.2 MODULES FOLLOWING BALLOTS-MARC

Following the production operation of the implemented BALLOTS-MARC module, each of the other ten modules described in section 2.3 will be implemented in turn. Figure 13 shows the screen formats, the files, and the inputs and outputs that will be added to the BALLOTS system with the addition of each module.



MODULE	SCREENS
1. B-MARC	FULL BIBLIO. DISPLAY BIBLIO. INPUT (MODIF.) GENERAL SCREEN (LOG) HOLDINGS (& CARD REQUEST) ORDERING SEARCH
2. IPF (IN PROCESS FILE)	FULL ACQUISITION DISPLAY BIBLIO. UPDATE HOLDINGS UPDATE ACQUISITION UPDATE BIBLIO./ACQUISITION/HOLDINGS DISPLAY MATRIX (SINGLE ITEM vs. STATUS)

MODULE	SCREENS
3. PO/ORG (PURCH. ORD. & ORIG. CAT.)	-----
4. NPO (NON PURCH. ORDER)	-----
5. CDF (CATALOG DATA FILE)	REFERENCE DISPLAY REFERENCE INPUT REFERENCE UPDATE
6. INV (MEYER INVENTORY FILE)	-----
7. BOOK (MEYER BOOK CATALOG)	-----
8. AC/AC (AUTO. CLAIM. & CANC.)	-----
9. CIRC (CIRCULATION)	CIRCULATION DISPLAY CIRCULATION INPUT
10. SO/OP (STAND. ORDER, OUT-OF-PRINT)	-----
11. RP (RESERVE PROCESSING)	RES. PRDC. DISPLAY

1. STANDING SEARCH REQ.
1. STATISTICAL AND MANAGEMENT REPORTS
2. CLAIM NOTICE - FORCE
2. CANCELLATION - FORCE
3. NPAC NOTICE
3. TITLE II SLIPS
4. INVOICE CLAIM - AUTO.
8. MATERIAL CLAIM - AUTO.
8. CANCELLATION - AUTO.
9. DATE DUE SLIPS
9. HOLD/RECALL NOTICES
9. FINE AND BOOK BILLS
10. STANDING ORDERS
10. SEARCH & QUOTE LETTER
11. RESERVE PROCESSING

Figure 13. BALLOTS--Features by Module

During the pilot operation of each module, the library analysts and library supervisors will monitor and audit the system continually. Debugging, fine-tuning to optimize the system, and minor modifications to make the system more convenient and useful to the library will also take place during pilot production. When the library is satisfied with a module, it will be considered in production status, allowing the analysts and programmers to concentrate on the next module. Since the time for acceptance is determined by the library and is therefore a development variable, the implementation dates for each module may vary from the schedule date. If the development work for a module is completed ahead of schedule, the library will have the opportunity to implement the module earlier than scheduled.

As can be seen in Figure 12, the parallel progress of tasks towards completing a module is such that one group may be free to turn to other work while another group is working on the module. Thus the library systems analysts used the period between BALLOTS-MARC requirements definition and users' manual preparation to compile the requirements documentation for the following module, MARC-IPF, and submit the document to the library for approval. Once a module is in production operation, the programmers can turn to the already documented and approved requirements they need in order to begin their work on the next module.

4.3 EXTENDING MODULES TO NETWORK LIBRARIES

For the first four to eight months after a module has been implemented and placed in production, it will be closely observed and monitored. During this period, the network version of the module will be checked out and tested for network use, and the network libraries will install equipment and conduct training classes and acceptance tests. When this is completed, the module will be put into network pilot production. Thus a module will have four to eight months of heavy production use in its original version prior to network installation. This will reduce implementation time and effort for the network users.

REFERENCES

1. Epstein, A. H., Douglas Ferguson, Eleanor Montague, John Schroeder, and Allen B. Veaner, BIBLIOGRAPHIC AUTOMATION OF LARGE LIBRARY OPERATIONS USING A TIME-SHARING SYSTEM: PHASE I (Stanford, California: Stanford University Libraries, April 1971).

2. Fasana, Paul J., and Allen B. Veaner, eds., COLLABORATIVE LIBRARY SYSTEMS DEVELOPMENT (Cambridge, Mass.: The M.I.T. Press, 1971).

3. Shoffner, Ralph M., "The Economics of National Automation of Libraries," LIBRARY TRENDS, April 1970, 18:448-63.

4. DESIGN OF THE STANFORD PUBLIC INFORMATION RETRIEVAL SYSTEM (SPIRES II), Volume I (Stanford, California: SPIRES/BALLOTS Project, July 1971).

5. Avram, Henriette D., THE MARC PILOT PROJECT: FINAL REPORT (Washington, D.C.: Library of Congress, 1968).

6. Swanson, Gerald L., "Selected Statistics Based on L.C. MARC Bibliographic Records," Systems Office, Columbia University Libraries, technical notes nos. 2 and 3, September and October 1970.

7. BALLOTS II SYSTEM REQUIREMENTS: DATA ELEMENTS NOTEBOOK, Volume 1, "Acquisition and Cataloging" (Project BALLOTS in-house documentation).

BIBLIOGRAPHY

DOCUMENTS RELATING TO PROJECT BALLOTS

This list includes selected reports submitted to the Office of Education.

ED numbered documents are available from LEASCO Information Products, Inc., P.O. Drawer O, Bethesda, Maryland 20014.

PB numbered documents are available from the National Technical Information Service, Operations Division, Springfield, Virginia 22151.

I PUBLICATIONS

AUTOMATION NEWSLETTER: PROJECT BALLOTS. Automation Division, Stanford University Libraries. No. 1, June 1969; No. 2, September 1969; No. 3-4 (combined issue), March 1970. <Available without charge from the SPIRES/BALLOTS Documentation Office, Stanford Computation Center, Stanford, California 94305.>

Epstein, A. H., Douglas Ferguson, and Eleanor Montague. "An On-Line Network--Cooperative Planning with Several Libraries." Paper read at the ASIS Conference, November 7-11, 1971, at Denver, Colorado; in 1971 ASIS PROCEEDINGS, pp. 227-31.

Epstein, A. H., and Allen B. Veaner. "The Management and Administration of Large-Scale System Development Programs." Paper read at the Special Libraries Association Conference, June 7, 1971, at San Francisco, California; in SPECIAL LIBRARIES (in press).

Parker, Edwin B. "Developing a Campus Based Information Retrieval System." In PROCEEDINGS, STANFORD CONFERENCE ON COLLABORATIVE LIBRARY SYSTEMS DEVELOPMENT (at Stanford University, Stanford, California, October 4-5, 1968). Stanford University Libraries, Stanford, California, 1969. Pp. 213-30. <ERIC document number ED 031 281; microfiche $1.00, hard copy $11.50.>

Veaner, Allen B. "The Application of Computers to Library Technical Processing." COLLEGE AND RESEARCH LIBRARIES, vol. 31, no. 2 (1970), pp. 36-42.

Veaner, Allen B. "Major Decision Points in Library Automation." Paper read at the 75th Annual Meeting of the Association of Research Libraries, January 17-18, 1970, at Chicago, Illinois. COLLEGE AND RESEARCH LIBRARIES, vol. 31, no. 5 (1970), pp. 299-312. The full text as delivered appears in the MINUTES of the Association's 75th Annual Meeting, pages 2-33.

Veaner, Allen B. "Stanford University Libraries, Project BALLOTS, A Summary." In PROCEEDINGS, STANFORD CONFERENCE ON COLLABORATIVE LIBRARY SYSTEMS DEVELOPMENT (at Stanford University, Stanford, California, October 4-5, 1968). Stanford University Libraries, Stanford, California, 1969. Pp. 42-49. <ERIC document number ED 031 281; microfiche $1.00, hard copy $11.50.>

Veaner, Allen B., A. H. Epstein, and John Schroeder. "The Stanford Experience." In Fasana, Paul J., and Allen B. Veaner, eds., COLLABORATIVE LIBRARY SYSTEMS DEVELOPMENT. Cambridge, Mass.: The M.I.T. Press, 1971. Pp. 42-68.

Weber, David C. "Personnel Aspects of Library Automation." Paper read at the Information Science and Automation Division Program Meeting of The American Library Association, June 28 - July 3, 1970, at Detroit, Michigan. JOURNAL OF LIBRARY AUTOMATION, vol. 4, no. 1 (1971), pp. 27-37.

Weber, David C. "Independent Development." In Fasana, Paul J., and Allen B. Veaner, eds., COLLABORATIVE LIBRARY SYSTEMS DEVELOPMENT. Cambridge, Mass.: The M.I.T. Press, 1971. Pp. 98-101.

II DOCUMENTS AND REPORTS

BALLOTS PROGRESS REPORT 3/27/69 - 6/26/69. SPIRES/BALLOTS Project, Stanford University, Stanford, California, July 1969. 76 pp. <ERIC document number ED 030 777; microfiche $1.50, hard copy $20.20.>

Ferguson, Douglas, ed. PROJECT CONTROL NOTEBOOK. 2nd ed., rev. SPIRES/BALLOTS Project, Stanford University, Stanford, California, December 1970. 180 pp.

Epstein, A. H., Douglas Ferguson, Eleanor Montague, John Schroeder, and Allen B. Veaner. FINAL REPORT PROJECT NO. 7-1145 GRANT NO. OEG-1-7-071145-4428 BIBLIOGRAPHIC AUTOMATION OF LARGE LIBRARY OPERATIONS USING A TIME-SHARING SYSTEM: PHASE I. Stanford University Libraries, Stanford, California, April 1971. 297 pp. (Without appendices, 101 pp.) <ERIC document number ED 049 786; microfiche $.65, hard copy $13.16. Also available from The LARC Association, P. O. Box 27235, Tempe, Arizona 85282, without appendices, as volume 4, issue 3 of THE LARC REPORTS.>

Ferguson, Douglas. "Information Retrieval (SPIRES) and Library Automation (BALLOTS) at Stanford University." SPIRES/BALLOTS Project, Stanford University, Stanford, California, November 1970. 12 pp. <ERIC document number ED 008 543; microfiche $.65, hard copy $3.28.>

Montague, Eleanor. SUMMARY OF A FEASIBILITY STUDY ON THE PARTICIPATION OF FOUR COLLEGES AND UNIVERSITIES IN A STANFORD

UNIVERSITY LIBRARY AUTOMATION NETWORK. SPIRES/BALLOTS Project, Stanford University, Stanford, California, November 1971. 78 pp. <Available from the SPIRES/BALLOTS Documentation Office, Stanford Computation Center, Stanford, California 94305, for $6.50 prepaid.>

SYSTEM SCOPE FOR LIBRARY AUTOMATION AND GENERALIZED INFORMATION STORAGE AND RETRIEVAL AT STANFORD UNIVERSITY. SPIRES/BALLOTS Project, Stanford University, Stanford, California, 1970. <ERIC document number ED 038 153. Available from the SPIRES/BALLOTS Documentation Office, Stanford Computation Center, Stanford, California 94305, for $7.50 prepaid. Also available from The LARC Association, P. O. Box 27235, Tempe, Arizona 85282, as volume 4, issue 4 of THE LARC REPORTS.>

III MAJOR PRESENTATIONS

Epstein, A.H. "Information Flow." American Society for Information Science, Los Angeles Chapter, Los Angeles, California, December 1970.

Epstein, A.H. "Information Flow Analysis." Canadian Society for Information Science, Ottawa, Canada, March 1971.

Epstein A.H. "Library Automation at Stanford." American Society for Information Science, San Francisco Bay Area Chapter, Palo Alto, California, November 1970.

Epstein, A.H. "Management of Large Library Computer Projects." University of California System Development Task Force, Berkeley, California, January 1971.

Epstein, A.H. "On-line Search Systems." Fall Meeting of the Association of Scientific Information Dissemination Centers, Rochester, New York, September 1970.

Epstein, A.H., Douglas Ferguson, and John Schroeder. "Status Report to the BALLOTS Advisory Committee." Los Angeles, California, January 1971.

Parker, Edwin B. "Potential Interrelationships Between Library and Other Mass Media Systems." Paper read at the Conference on Interlibrary Communications and Information Networks, September 28 - October 2, 1970, at Warrenton, Virginia.

IV ABOUT BALLOTS (AND SPIRES)

Burns, Robert W., Jr. Review of System Scope document. COLLEGE AND RESEARCH LIBRARIES, May 1971, vol. 32, pp. 236-38.

Description of BALLOTS in "Directory of Scholars Active" section of COMPUTERS AND THE HUMANITIES, May 1971, P. 291.

Klinger, Karen. "Stanford Library Turns to Computer." SAN JOSE MERCURY, June 10, 1971.

"Libraries Seek University-Wide Computer Information Service." <STANFORD> CAMPUS REPORT, January 14, 1970, p. 7.

"Library Goal: Computerized Information Retrieval System." STANFORD OBSERVER, January 1970.

"Planning for a Stanford Library Automation Network." Stanford Computation Center Bulletin, August 18, 1971, vol. VII, no. 12, P. 4.

"Project BALLOTS: Progress Report." JOLA TECHNICAL COMMUNICATIONS, no. 5 (1970), p. 2.

A report on automation plans (SPIRES and BALLOTS) at Stanford. COLLEGE AND RESEARCH LIBRARIES NEWS, March 1970, p. 83.

V FILM

SPIRES/BALLOTS REPORT. Department of Communications, Stanford University, Stanford, California, 1969. <A 15-minute, color, 16mm film giving an overview of the library automation and information retrieval problem in general and of Stanford's approach to it. Written and directed by D. B. Jones. Copies may be rented from Extension Media Center, University of California, Berkeley 94720. Rental charge, $15.00 for 24 hours; purchase Price, $180.00.>

APPENDIX A

RELATED RESEARCH: AN EXCERPT FROM THE CLSD FINAL REPORT

"The CLSD project, if viewed within the context of the original objectives and the perspective of the emerging state-of-the-art characteristic of the late 60's, was an extremely beneficial experience for those who participated, and to a lesser degree, the general library community. Though the findings of the project were often negative, these facts in themselves are important because they helped the participants (and, one hopes, to some degree, the general library community) to distinguish between the practical realities of library systems development and the popular misconceptions of the totally automated library.

"Based on the experience gained in the CLSD project, the following observations are offered:

a. Loosely defined, cooperative projects, such as CLSD, though beneficial for those participating, are not the best or even, perhaps, the most efficient ways of fostering collaborative work.

b. Cooperative work would have had more chance to be successful if done on a local level. The logistical problems of convening personnel from widely separated institutions are formidable and time consuming.

c. Cooperative work needs a strong commitment on both a technical and administrative level from all participants. The nature of the commitment must be clearly stated and fully understood.

d. Coordination of local development and work schedules is difficult to achieve. Detailed coordination of schedules is probably impossible to achieve among several independent projects. A truly joint effort probably requires that a single work schedule be developed and adhered to. This is probably possible only where there is agreement among the participants that a single program package will be developed and all local processing systems will use either a centralized processing facility or radically redesign local procedures to conform to the requirements of the program package.

e. Rapid communication and detailed documentation is vital to a systems project. There is no simple or practical method of insuring that communication and documentation is done effectively within a single institution. The problem is enormously aggravated in an inter-institutional effort.

f. Computer hardware configurations have played and probably will continue to play an important and decisive role in the design of library systems. Large scale, compatible computer systems for libraries are not yet technically feasible.

g. Standards are lacking in many areas of library systems work. Until standards are established, library systems design efforts will continue to result in unique processing systems reflecting the peculiarities of local environments, thereby reducing the possibility of developing transferable systems.

"Large scale library systems work is just now begining to emerge from a period of pioneering and research. In the next several years, practical, production-engineered systems will begin to be made available and used. The cost of developing such systems is enormous; therefore, few individual libraries will be able to afford to develop systems for themselves. The feasibility of libraries being able to cooperatively develop large scale, complex, transferable systems on an informal basis is dubious. At present, there are other, more promising approaches; these include:

1) local, joint efforts for the cooperative development of similar and compatible systems wherein participating libraries are financially and administratively committed;

2) cooperative effort for the development of a single, central or regional system wherein, again, participating libraries are financially and administratively committed;

3) centralized development of a general-purpose library package, with options for customer modification."

The CLSD report acknowledges the problems associated with each of these three approaches to development, and in particular suggests that although generalized systems exact a certain penalty in reduced efficiency, they facilitate networks by preserving enough local autonomy to ensure political viability. One thing about networks is quite clear in the United States: no network can dictate its design and services to its intended clientele. It is for this reason that a strenuous user-involvement effort has been undertaken in CLAN--to make certain that the available applications work to the satisfaction of their users.

(Taken from: COLLABORATIVE PROGRAM IN LIBRARY SYSTEM DEVELOPMENT: FINAL REPORT, A Report to the National Science Foundation, NSF-GN-724, July 1971, pp. 19-20.)

APPENDIX B

SAMPLE PAGES FROM BALLOTS
SYSTEM REQUIREMENTS DOCUMENTATION

9/17/70

THE ORDERING PROCESS

REQUESTS

Material is acquired for the Stanford University Libraries by two prinicpal methods: (1) generating a purchase order in response to a specific request for a title, and (2) receiving a physical volume through an approval or blanket order plan, from an overseas buying trip, by exchange, or as a gift.

The ordering process handles the acquisition of specific material by purchase order. Organizationally, the ordering process is centrally coordinated by the Stanford University Libraries Acquisition Department Order Division.

Requests arrive for order processing from faculty, staff, students, library departments, Coordinate libraries, and other users of the centralized ordering process. The process provides three services to its users: (1) pre-ordering searching, if required, (2) creation of an In Process File record for a title to be ordered, and (3) production from the In Process File record of all documentary outputs required in ordering.

SEARCHING

Requests for which pre-order searching is not required are immediately keyed into the In Process File. All other requests are searched for before a purchase order is generated. The purpose of this search is to:

(1) Verify that the requested title is not an unneeded duplicate of a title currently held, in process, or expected on approval.

(2) Verify the bibliographic description of the title in order to communicate accurate information to the vendor on the purchase order.

(3) Locate Library of Congress bibliographic information for new acquisitions to communicate accurate information to the vendor on the purchase order and to speed cataloging once the material is received.

ON-LINE FILES

In addition to the standard printed reference tools, catalogs, and Stanford's manual files, four on-line computer files are available for searching:

(1) The In Process File - the central control file for all titles in process.

(2) The MARC File - the Stanford file of Library of Congress MARC records.

(3) The Meyer Inventory File - the central file of bibliographic and circulation records for the Meyer Memorial Library.

(4) The Catalog Data File - a central file of bibliographic records for titles processed through the automated Cataloging Subsystem (not including Meyer titles).

These files are searched at a cathode ray tube (CRT) visual display terminal. Bibliographic data can be copied from the MARC, Meyer, and Catalog Data Files into the In Process File.

IN PROCESS FILE RECORDS

The In Process File contains bibliographic and acquisition data for all titles on order. A title can be acquired by different methods (e.g., gift, purchase order, approval), from different sources, at different times. A single bibliographic record may have associated with it several separate acquisition records. Each bibliographic record has an acquisition record for each order of the same title.

An In Process File record is created in three ways. During pre-order searching, if bibliographic information is found for the title in MARC, Meyer, or the Catalog Data File, the information is transferred to the In Process File, coupled with new acquisition information that has been keyed at a CRT, to form a new record. If bibliographic information is already in the In Process File, only newly keyed acquisition information is added to form the new record. If no machine-readable data is found, both bibliographic and acquisition data are keyed at the terminal to create a new In Process File record.

DOCUMENTARY OUTPUTS

When the In Process File is updated with a new acquisiton record, information for all ordering documentary outputs required for the title is extracted and sent to the output print program for overnight sorting, formatting, and printing. Ordering outputs are:

(1) Purchase Order - sent to the vendor to order a specific title.

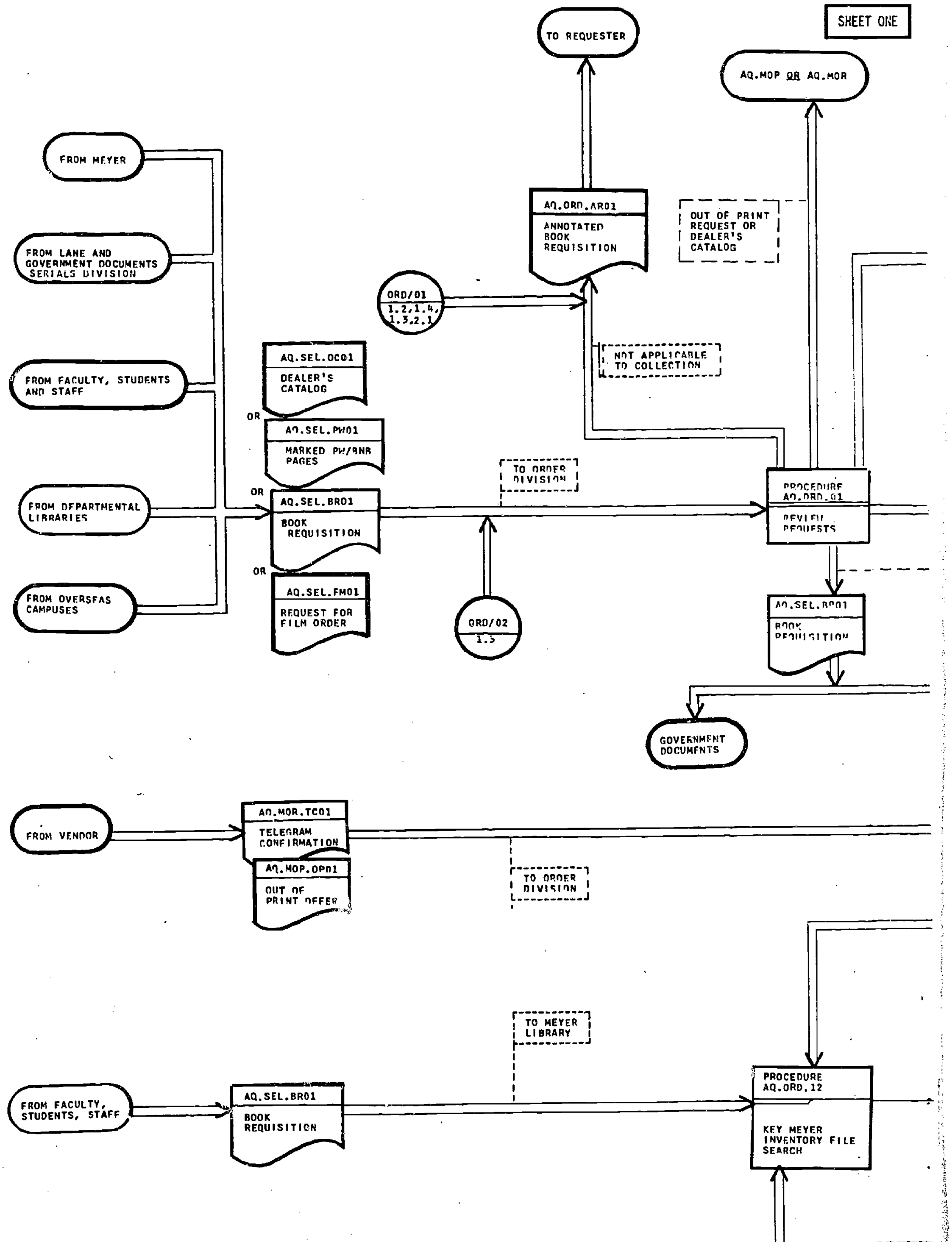
(2) Dealer Report - sent with the purchase order to be returned by the vendor with the material, or used as a report if the order cannot be filled.

(3) Abel Accounts Receivable Slip - sent with the purchase order if Abel is the vendor. Used for Abel's internal processing.

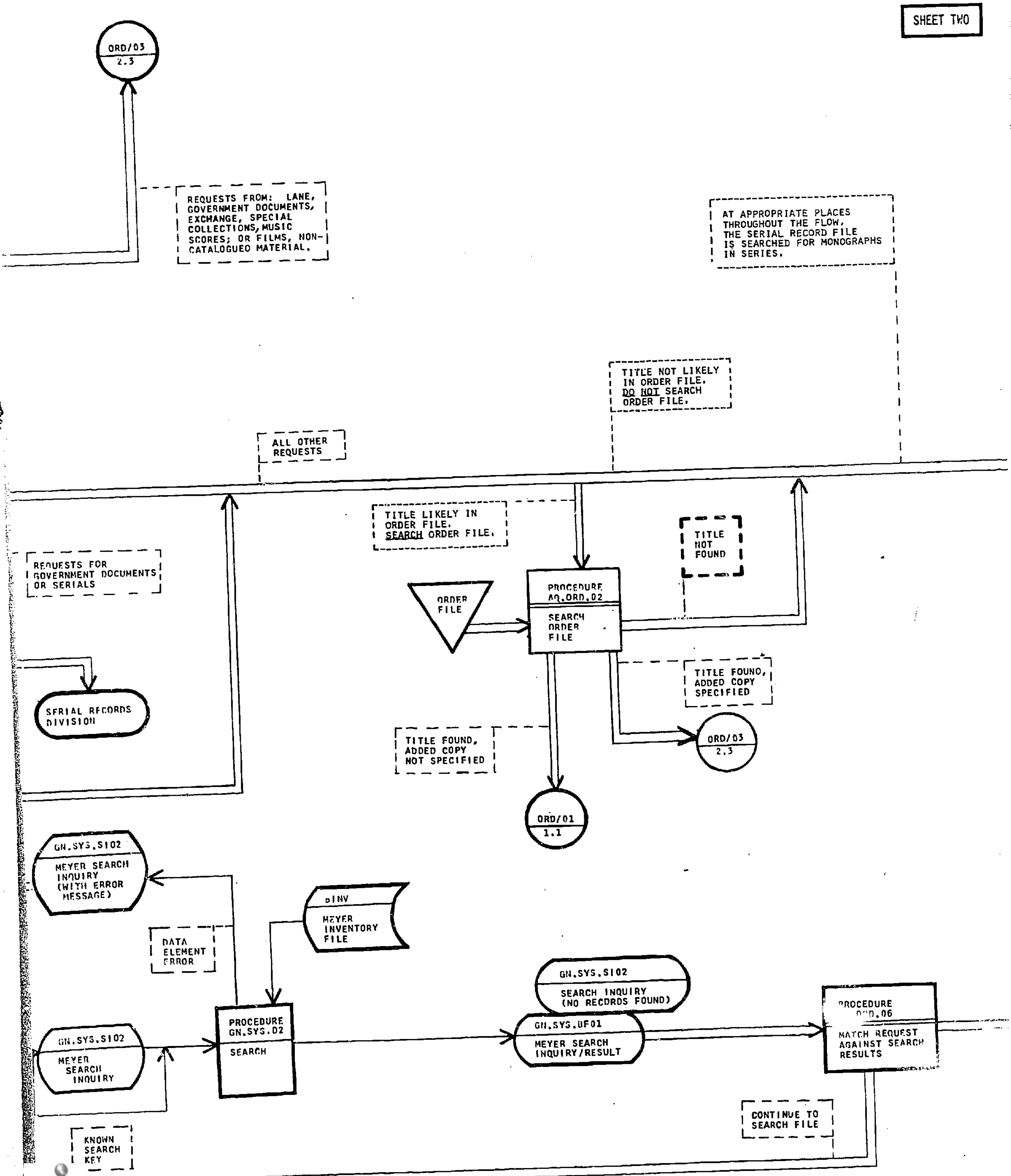
(4) Abel Original Invoice - sent with the purchase order if Abel is the vendor. Returned with the material as an original invoice.

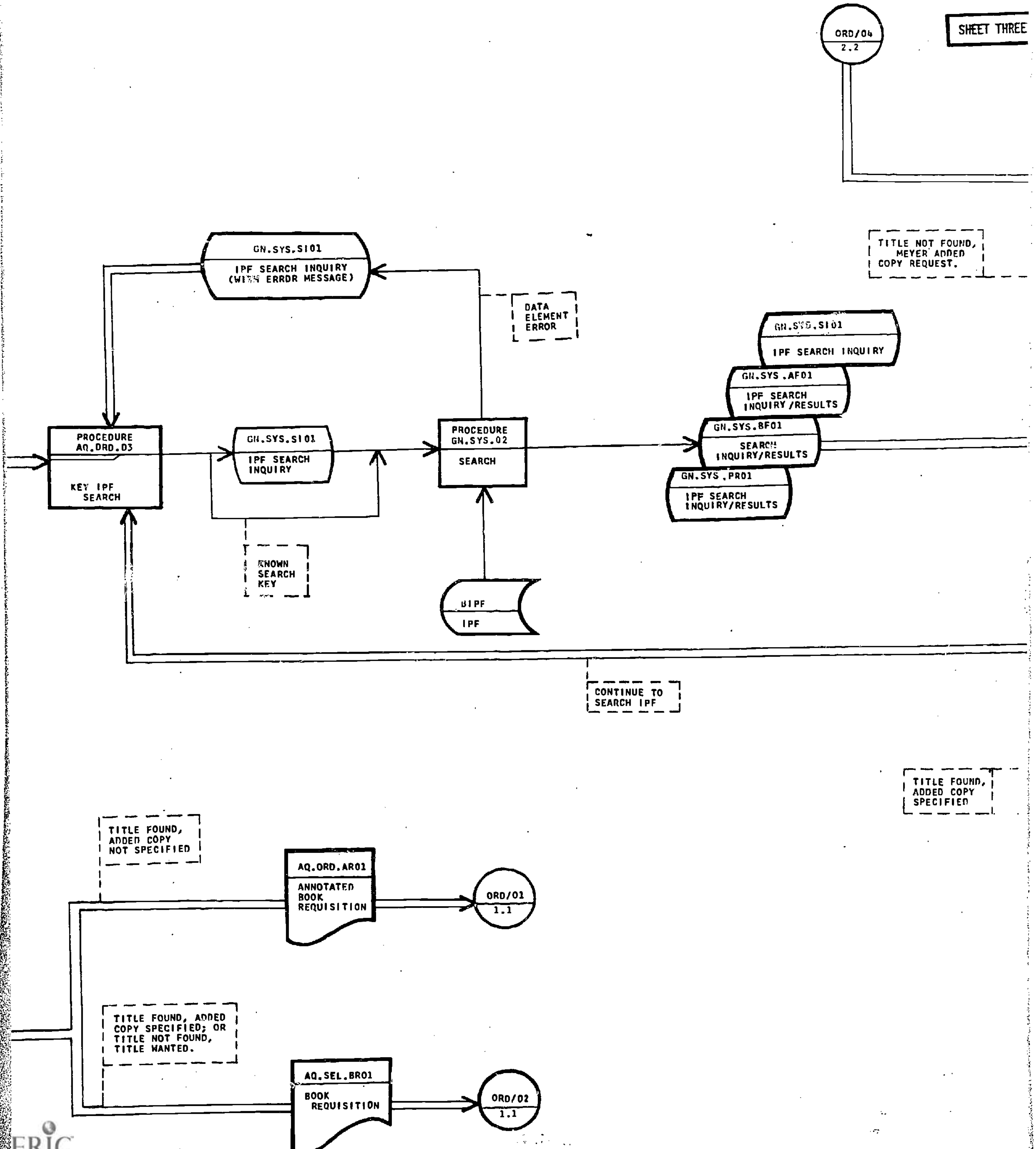
(5) Process Slips - sent to a user of the ordering process who maintains a manual file for his own internal processing.

(6) NPAC Notice - sent to the Library of Congress for each title in the scope of the National Program for Acquisitions and Cataloging for which no Library of Congress bibliographic data was found during the ordering process.

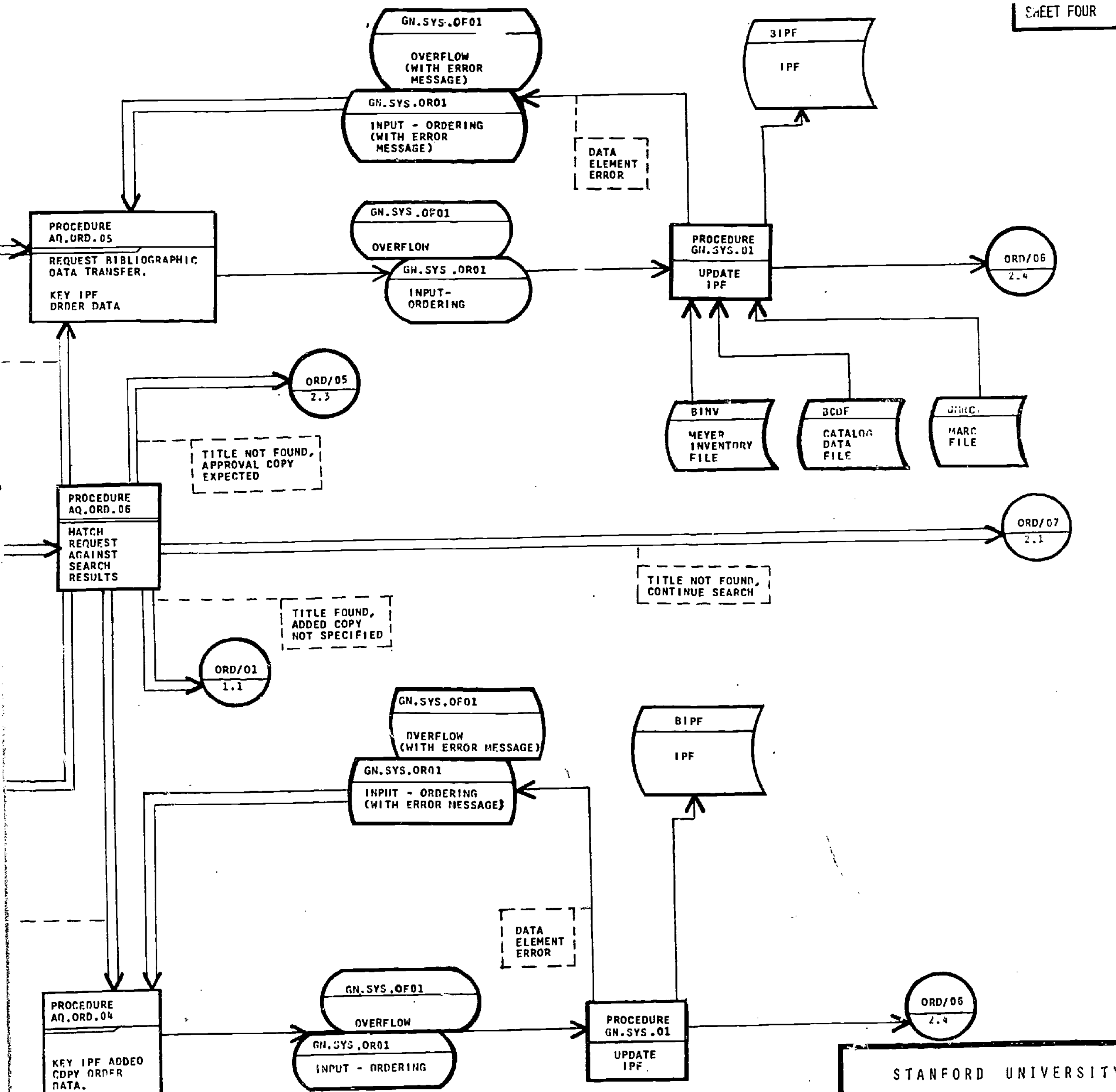


SHEET ONE
TO REQUESTER
AQ.MOP OR AQ.MOR
FROM MEYER
FROM LANE AND GOVERNMENT DOCUMENTS SERIALS DIVISION
FROM FACULTY, STUDENTS AND STAFF
FROM DEPARTMENTAL LIBRARIES
FROM OVERSEAS CAMPUSES
AQ.ORD.AR01
ANNOTATED BOOK REQUISITION
OUT OF PRINT REQUEST OR DEALER'S CATALOG
ORD/01
1.2,1.4, 1.3.2.1
NOT APPLICABLE TO COLLECTION
AQ.SEL.OC01
DEALER'S CATALOG
OR
AQ.SEL.PW01
MARKED PW/BNB PAGES
OR
AQ.SEL.BR01
BOOK REQUISITION
OR
AQ.SEL.FM01
REQUEST FOR FILM ORDER
TO ORDER DIVISION
ORD/02
PROCEDURE AQ.ORD.01
REVIEW REQUESTS
AQ.SEL.BR01
BOOK REQUISITION
GOVERNMENT DOCUMENTS
FROM VENDOR
AQ.MOR.TC01
TELEGRAM CONFIRMATION
AQ.MOP.OP01
OUT OF PRINT OFFER
TO ORDER DIVISION
TO MEYER LIBRARY
FROM FACULTY, STUDENTS, STAFF
AQ.SEL.BR01
BOOK REQUISITION
PROCEDURE AQ.ORD.12
KEY MEYER INVENTORY FILE SEARCH


SHEET TWO
ORD/03
2.3
REQUESTS FROM: LANE, GOVERNMENT DOCUMENTS, EXCHANGE, SPECIAL COLLECTIONS, MUSIC SCORES; OR FILMS, NON-CATALOGUED MATERIAL.
AT APPROPRIATE PLACES THROUGHOUT THE FLOW, THE SERIAL RECORD FILE IS SEARCHED FOR MONOGRAPHS IN SERIES.
TITLE NOT LIKELY IN ORDER FILE. DO NOT SEARCH ORDER FILE.
ALL OTHER REQUESTS
TITLE LIKELY IN ORDER FILE. SEARCH ORDER FILE.
TITLE NOT FOUND
REQUESTS FOR GOVERNMENT DOCUMENTS OR SERIALS
ORDER FILE
PROCEDURE AQ.ORD.02
SEARCH ORDER FILE
SERIAL RECORDS DIVISION
TITLE FOUND, ADDED COPY SPECIFIED
TITLE FOUND, ADDED COPY NOT SPECIFIED
ORD/03
2.3
ORD/01
1.1
GN.SYS.S102
MEYER SEARCH INQUIRY (WITH ERROR MESSAGE)
INV
MEYER INVENTORY FILE
DATA ELEMENT ERROR
GN.SYS.S102
SEARCH INQUIRY (NO RECORDS FOUND)
GN.SYS.S102
MEYER SEARCH INQUIRY
PROCEDURE GN.SYS.D2
SEARCH
GN.SYS.UF01
MEYER SEARCH INQUIRY/RESULT
MATCH REQUEST AGAINST SEARCH RESULTS
CONTINUE TO SEARCH FILE
KNOWN SEARCH KEY



ORD/04
2.2
SHEET THREE
TITLE NOT FOUND, MEYER ADDED COPY REQUEST.
GN.SYS.SI01
IPF SEARCH INQUIRY (WITH ERROR MESSAGE)
DATA ELEMENT ERROR
GN.SYS.SI01
IPF SEARCH INQUIRY
GN.SYS.AF01
IPF SEARCH INQUIRY/RESULTS
GN.SYS.BF01
SEARCH INQUIRY/RESULTS
GN.SYS.PR01
IPF SEARCH INQUIRY/RESULTS
PROCEDURE AQ.ORD.03
KEY IPF SEARCH
GN.SYS.SI01
IPF SEARCH INQUIRY
PROCEDURE GN.SYS.02
SEARCH
KNOWN SEARCH KEY
BIPF
IPF
CONTINUE TO SEARCH IPF
TITLE FOUND, ADDED COPY SPECIFIED
TITLE FOUND, ADDED COPY NOT SPECIFIED
AQ.ORD.AR01
ANNOTATED BOOK REQUISITION
ORD/01
1.1
TITLE FOUND, ADDED COPY SPECIFIED; OR TITLE NOT FOUND, TITLE WANTED.
AQ.SEL.BR01
BOOK REQUISITION
ORD/02
1.1



STANFORD UNIVERSITY LIBRARIES

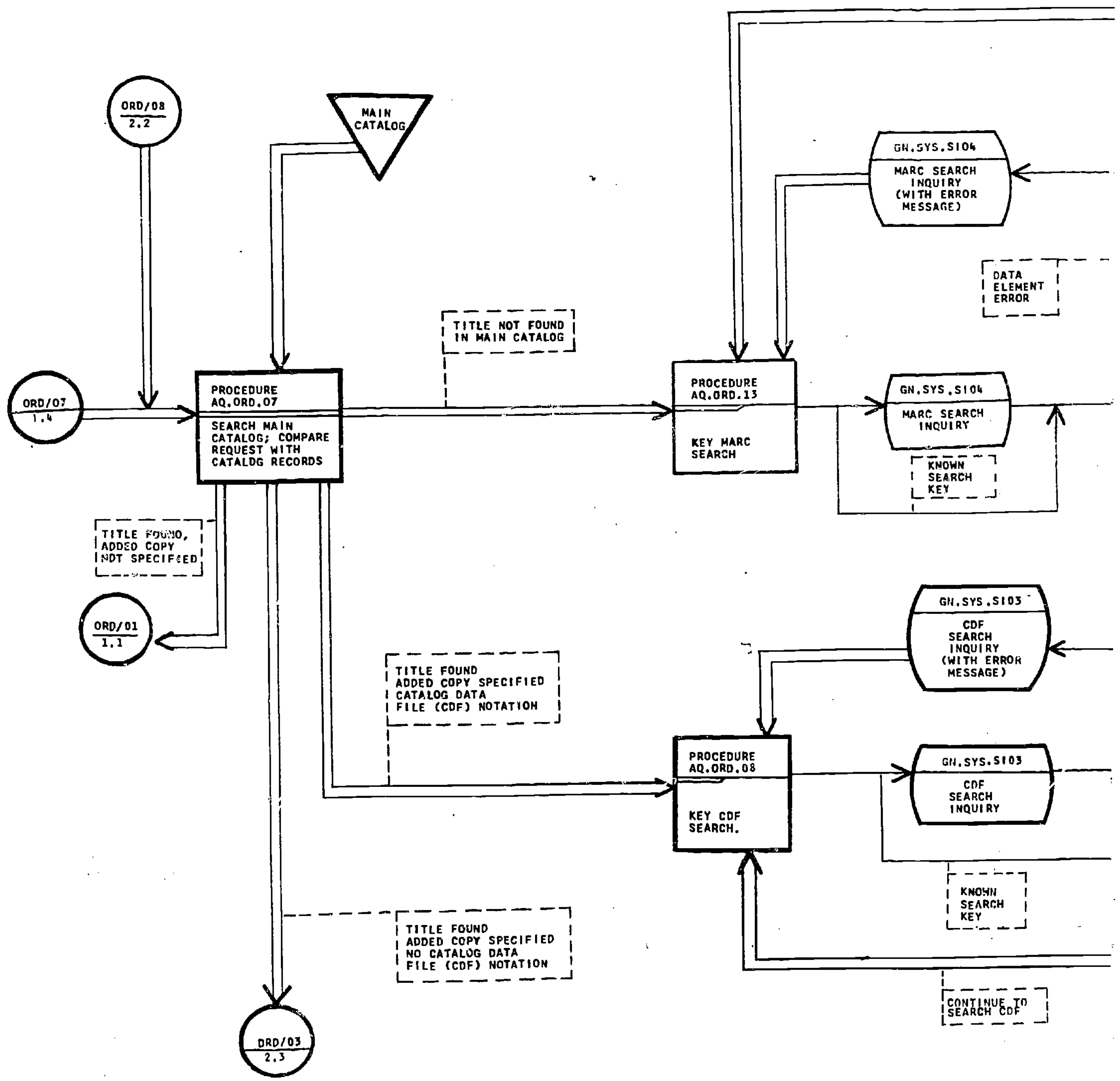
APPROVAL(s) REQUIRED: RALPH HANSEN, FRED LYNDEN

REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEPT INIT
1	9/17	ADD UPDATE NUMBER CHANGES			
0	7/17	INITIAL VERSION	EM		

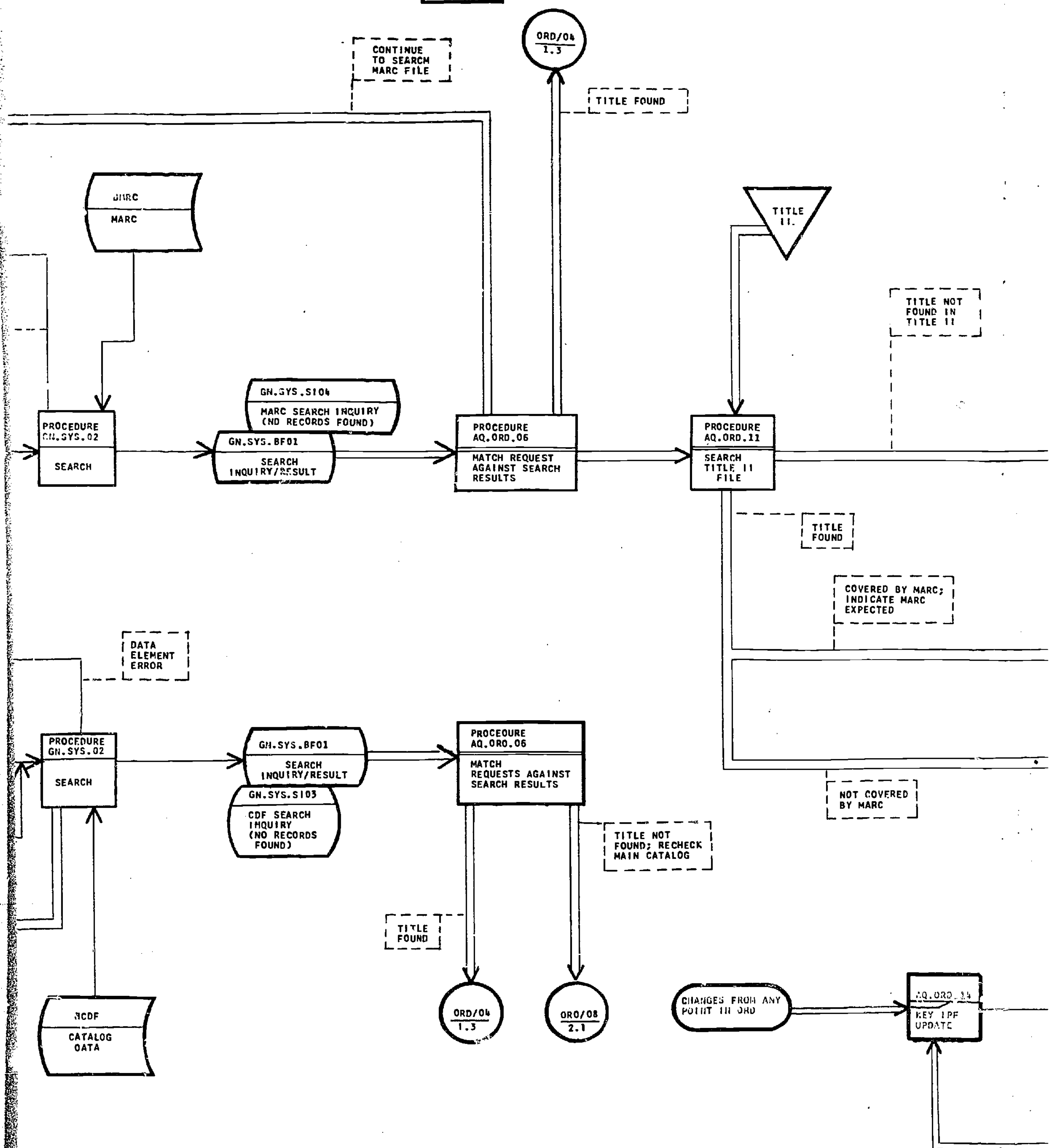
TITLE: ORDERING PROCESS ACQUISITION

ID: AQ.ORD 4 SHEETS PAGE 1 OF 2

SHEET ONE

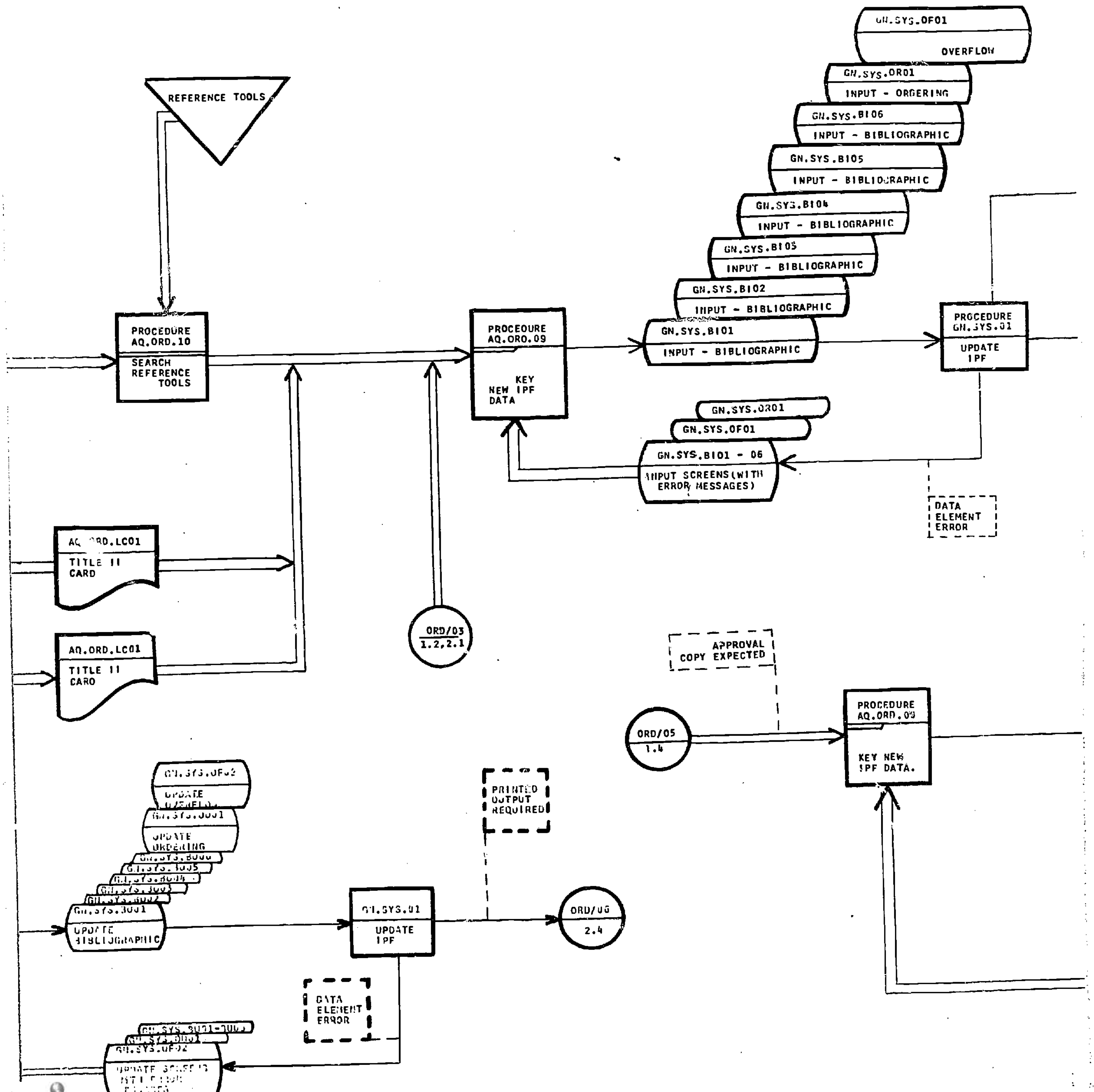

ORD/08
2.2
MAIN CATALOG
GN.SYS.S104
MARC SEARCH INQUIRY (WITH ERROR MESSAGE)
DATA ELEMENT ERROR
TITLE NOT FOUND IN MAIN CATALOG
ORD/07
1.4
PROCEDURE AQ.ORD.07
SEARCH MAIN CATALOG; COMPARE REQUEST WITH CATALOG RECORDS
PROCEDURE AQ.ORD.13
KEY MARC SEARCH
GN.SYS.S104
MARC SEARCH INQUIRY
KNOWN SEARCH KEY
TITLE FOUND, ADDED COPY NOT SPECIFIED
ORD/01
1.1
GN.SYS.S103
CDF SEARCH INQUIRY (WITH ERROR MESSAGE)
TITLE FOUND ADDED COPY SPECIFIED CATALOG DATA FILE (CDF) NOTATION
PROCEDURE AQ.ORD.08
KEY CDF SEARCH.
GN.SYS.S103
CDF SEARCH INQUIRY
KNOWN SEARCH KEY
TITLE FOUND ADDED COPY SPECIFIED NO CATALOG DATA FILE (CDF) NOTATION
CONTINUE TO SEARCH CDF
ORD/03
2.3

SHEET TWO



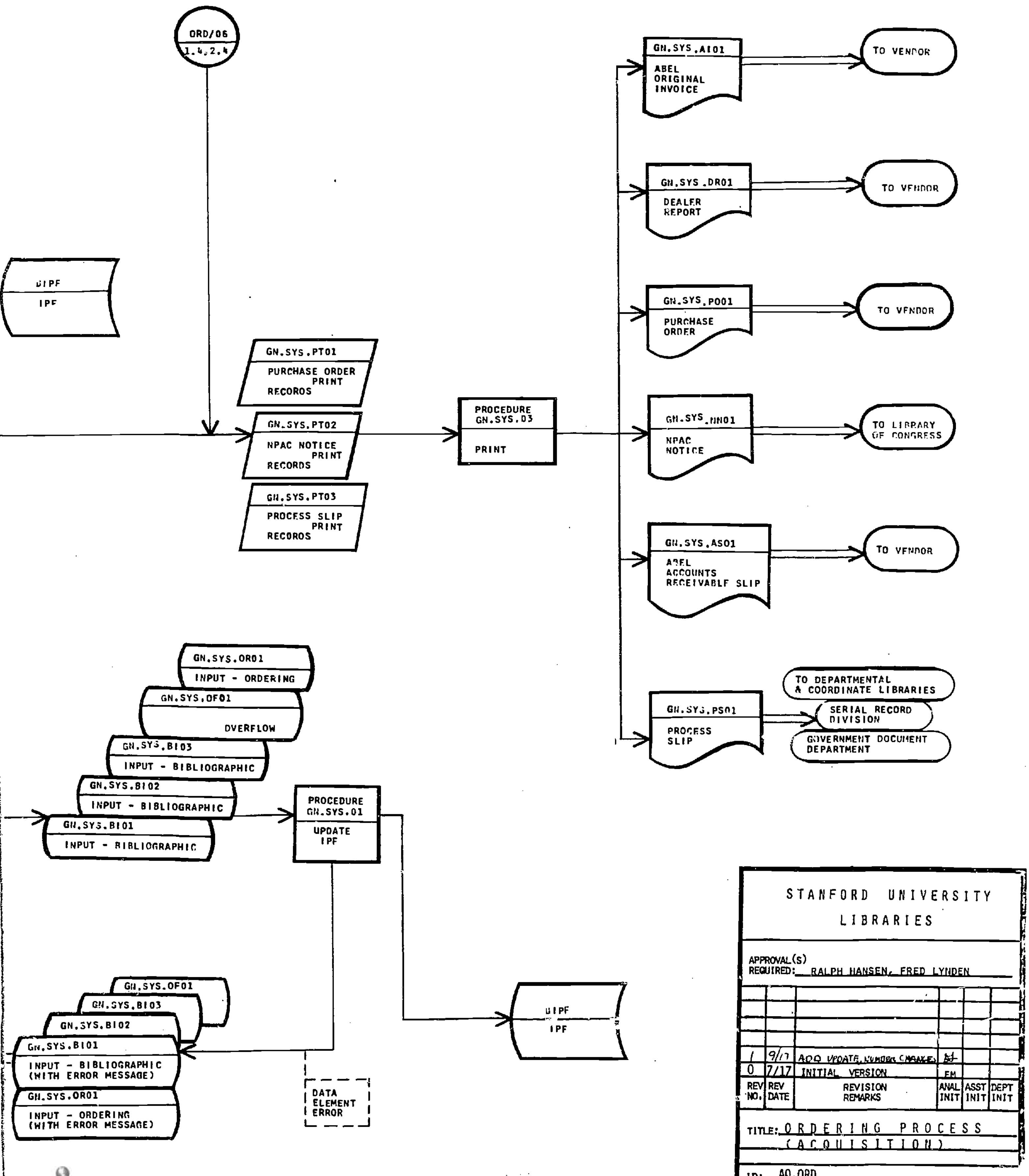
CONTINUE TO SEARCH MARC FILE
ORD/04 1.3
TITLE FOUND
MARC
TITLE II
TITLE NOT FOUND IN TITLE II
GN.SYS.S104
MARC SEARCH INQUIRY (NO RECORDS FOUND)
PROCEDURE GN.SYS.02
SEARCH
GN.SYS.BF01
SEARCH INQUIRY/RESULT
PROCEDURE AQ.ORD.06
MATCH REQUEST AGAINST SEARCH RESULTS
PROCEDURE AQ.ORD.11
SEARCH TITLE II FILE
TITLE FOUND
COVERED BY MARC; INDICATE MARC EXPECTED
DATA ELEMENT ERROR
PROCEDURE GN.SYS.02
SEARCH
GN.SYS.BF01
SEARCH INQUIRY/RESULT
GN.SYS.S103
CDF SEARCH INQUIRY (NO RECORDS FOUND)
PROCEDURE AQ.ORD.06
MATCH REQUESTS AGAINST SEARCH RESULTS
NOT COVERED BY MARC
TITLE NOT FOUND; RECHECK MAIN CATALOG
TITLE FOUND
ORD/04 1.3
ORD/08 2.1
CHANGES FROM ANY POINT IN ORD
KEY IPF UPDATE
CATALOG DATA

SHEET THREE



GN.SYS.OF01
OVERFLOW
GN.SYS.OR01
INPUT - ORDERING
GN.SYS.BI06
INPUT - BIBLIOGRAPHIC
GN.SYS.BI05
INPUT - BIBLIOGRAPHIC
GN.SYS.BI04
INPUT - BIBLIOGRAPHIC
GN.SYS.BI03
INPUT - BIBLIOGRAPHIC
GN.SYS.BI02
INPUT - BIBLIOGRAPHIC
GN.SYS.BI01
INPUT - BIBLIOGRAPHIC
REFERENCE TOOLS
PROCEDURE AQ.ORD.10
SEARCH REFERENCE TOOLS
PROCEDURE AQ.ORD.09
KEY NEW IPF DATA
PROCEDURE GN.SYS.01
UPDATE IPF
GN.SYS.OR01
GN.SYS.OF01
GN.SYS.BI01 - 06
INPUT SCREENS (WITH ERROR MESSAGES)
DATA ELEMENT ERROR
AQ.ORD.LC01
TITLE II CARD
AQ.ORD.LC01
TITLE II CARD
ORD/03
1.2, 2.1
APPROVAL COPY EXPECTED
ORD/05
1.4
PROCEDURE AQ.ORD.09
KEY NEW IPF DATA.
PRINTED OUTPUT REQUIRED
GN.SYS.01
UPDATE IPF
ORD/06
2.4
UPDATE BIBLIOGRAPHIC
DATA ELEMENT ERROR

SHEET FOUR


ORD/06
1.4,2.4
GN.SYS.AI01
ABEL ORIGINAL INVOICE
TO VENDOR
GN.SYS.DR01
DEALER REPORT
TO VENDOR
GN.SYS.PO01
PURCHASE ORDER
TO VENDOR
UIPF
IPF
GN.SYS.PT01
PURCHASE ORDER PRINT RECOROS
GN.SYS.PT02
NPAC NOTICE PRINT RECORDS
GN.SYS.PT03
PROCESS SLIP PRINT RECOROS
PROCEDURE GN.SYS.03
PRINT
GN.SYS.NN01
NPAC NOTICE
TO LIBRARY OF CONGRESS
GN.SYS.AS01
ABEL ACCOUNTS RECEIVABLE SLIP
TO VENDOR
GN.SYS.OR01
INPUT - ORDERING
GN.SYS.OF01
OVERFLOW
GN.SYS.BI03
INPUT - BIBLIOGRAPHIC
GN.SYS.BI02
INPUT - BIBLIOGRAPHIC
GN.SYS.BI01
INPUT - BIBLIOGRAPHIC
PROCEDURE GN.SYS.01
UPDATE IPF
GN.SYS.PS01
PROCESS SLIP
TO DEPARTMENTAL & COORDINATE LIBRARIES
SERIAL RECORD DIVISION
GOVERNMENT DOCUMENT DEPARTMENT
GN.SYS.OF01
GN.SYS.BI03
GN.SYS.BI02
GN.SYS.BI01
INPUT - BIBLIOGRAPHIC (WITH ERROR MESSAGE)
GN.SYS.OR01
INPUT - ORDERING (WITH ERROR MESSAGE)
DATA ELEMENT ERROR
UIPF
IPF
STANFORD UNIVERSITY LIBRARIES
APPROVAL(S) REQUIRED: RALPH HANSEN, FRED LYNDEN
1 9/17 ADD UPDATE, NUMBER CHANGE
0 7/17 INITIAL VERSION FM
REV NO. REV DATE REVISION REMARKS ANAL INIT ASST INIT DEPT INIT
TITLE: ORDERING PROCESS (ACQUISITION)
ID: AQ.ORD
4 SHEETS PAGE 2 OF 2

PURCHASE ORDER MATERIAL RECEIPT PROCESS

Materials (physical volumes) here included are those that are received in the Acquisition Department Order Division (1) accompanied by a dealer report slip, (2) represented by a purchase order, or (3) routed from Serial Division check-in.

The only bibliographic searching performed in the purchase order material receipt process is as follows:

1. MARC search for a monograph in a series on standing order that is to be classed separately (not done in the current system);

2. Search of the National Union Catalog and reference tools to establish more accurate entry for certain Slavic materials.

The In Process File and the MARC file are the only files used in this process. The bibliographic or acquisition information contained in an IPF record may require other updating at the time of receipt besides the addition of material receipt information. New IPF records may be created for a series on standing order prior to automation that is to be classed separately, and for an individual monograph belonging to such a series. Bibliographic data may be copied into the In Process File from the MARC file.

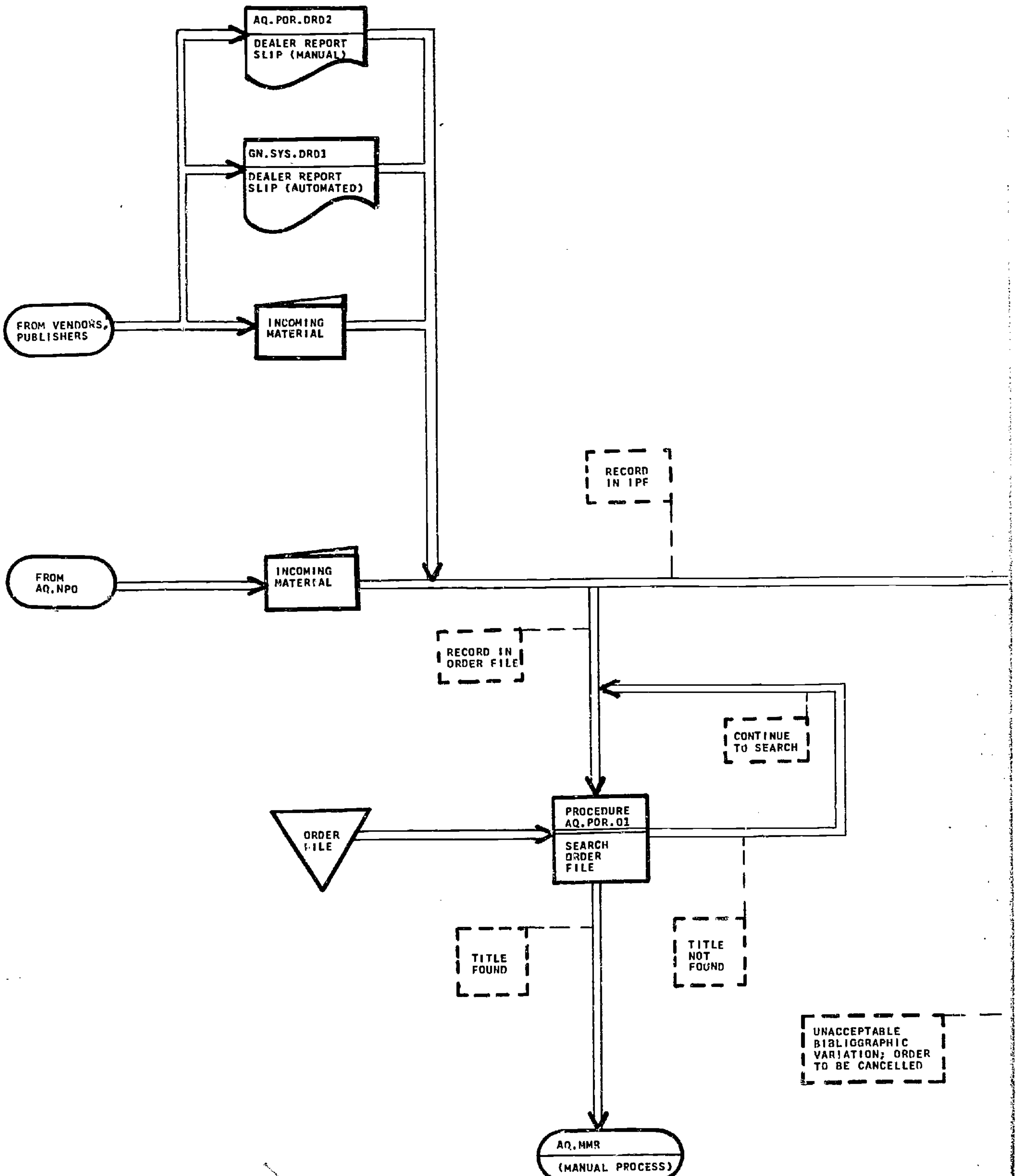
Three printed outputs are produced as a result of the purchase order material receipt process:

1. Catalog Data Slip--contains bibliographic data, shelving location, and other information needed by the cataloger; inserted in the material for routing to the Catalog Department.

2. Title II Slip--inserted in the material and sent to the Catalog Department for filing in the Title II File if Library of Congress catalog data in manual form is expected.

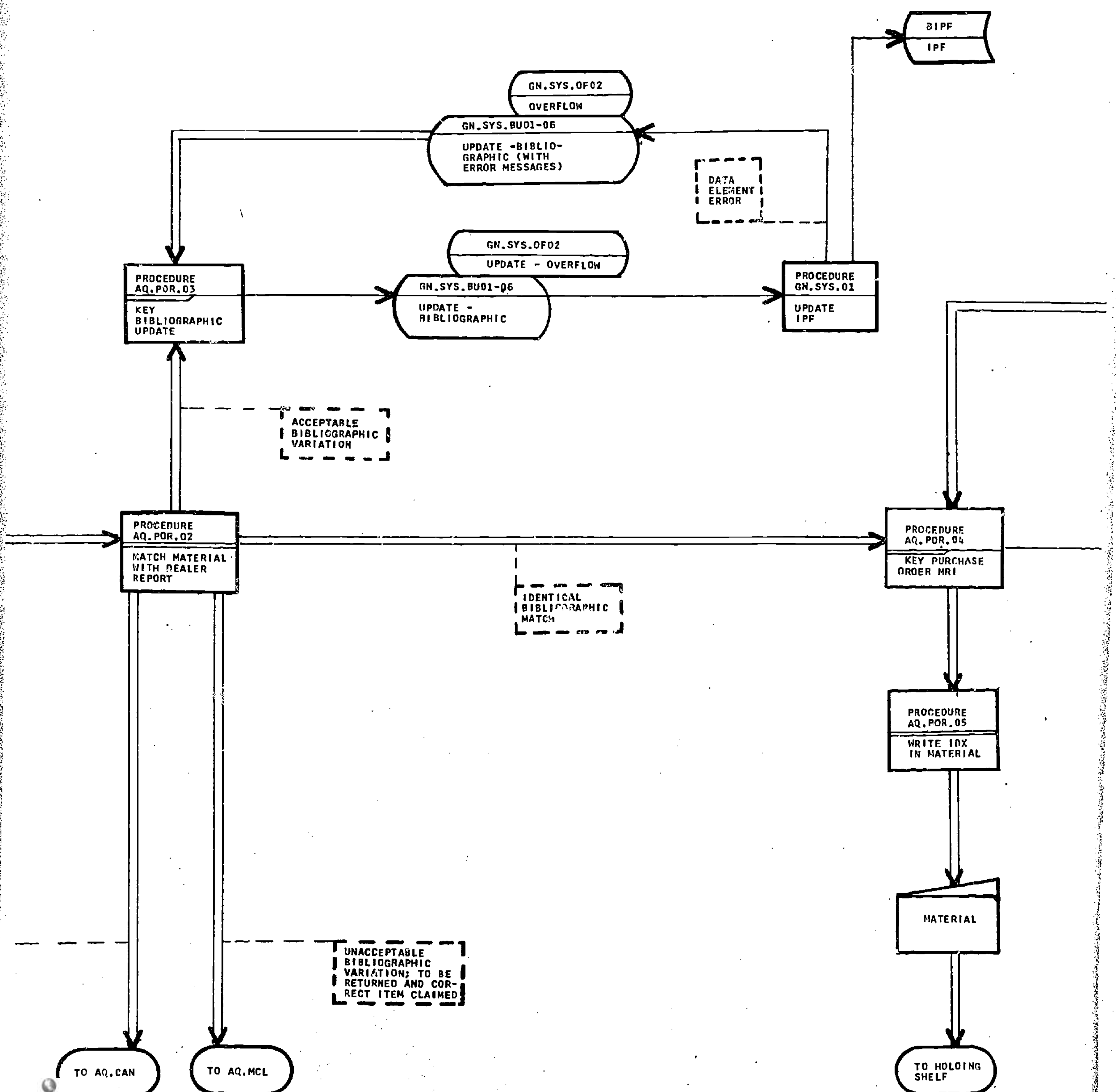
3. Requester Notice--sent to requester to notify him that requested material has been received.

SHEET ONE

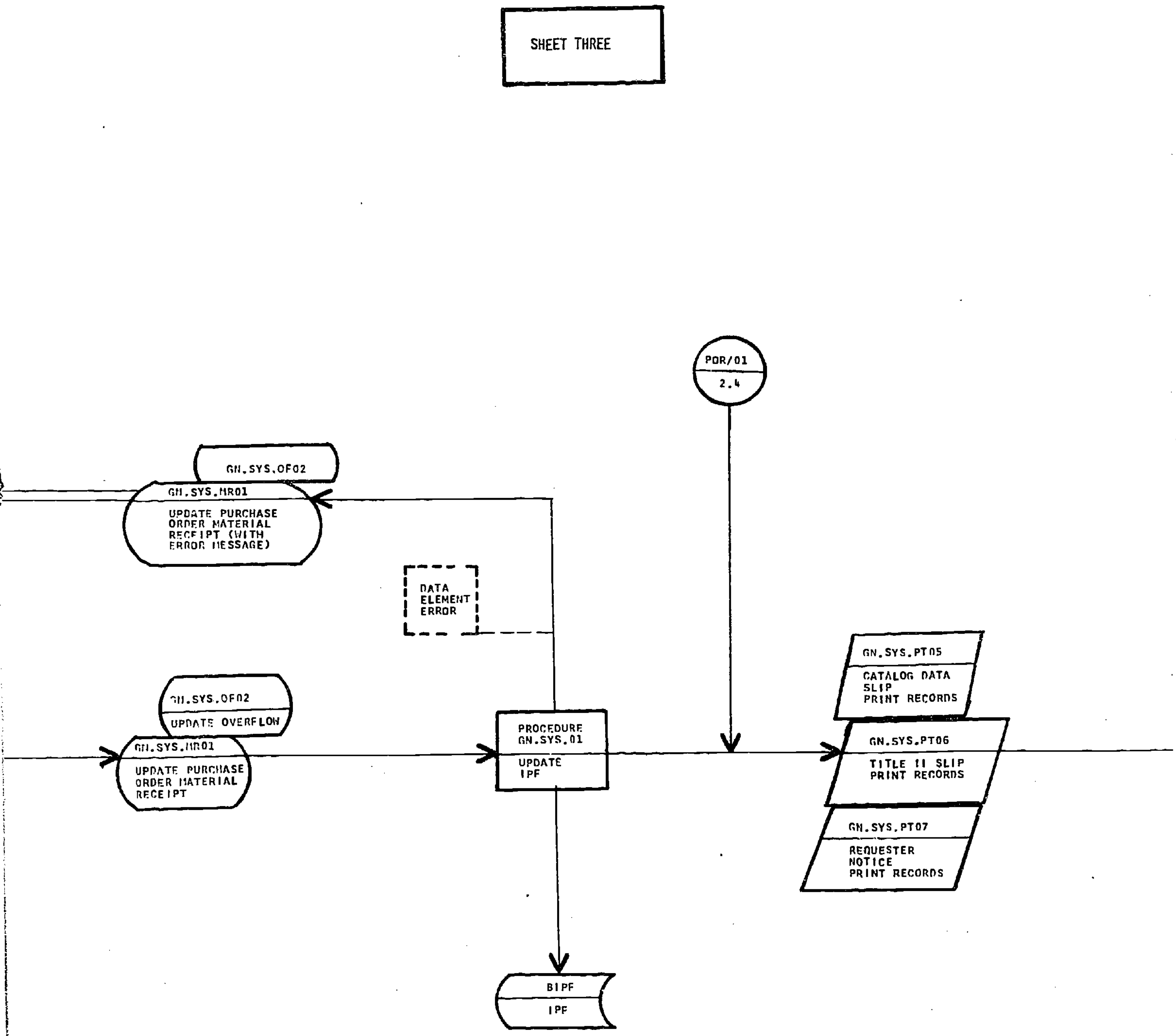


AQ.POR.DRD2
DEALER REPORT SLIP (MANUAL)
GN.SYS.DRD1
DEALER REPORT SLIP (AUTOMATED)
FROM VENDORS, PUBLISHERS
INCOMING MATERIAL
RECORD IN IPF
FROM AQ.NPO
INCOMING MATERIAL
RECORD IN ORDER FILE
CONTINUE TO SEARCH
ORDER FILE
PROCEDURE AQ.POR.01
SEARCH ORDER FILE
TITLE FOUND
TITLE NOT FOUND
UNACCEPTABLE BIBLIOGRAPHIC VARIATION; ORDER TO BE CANCELLED
AQ.MMR
(MANUAL PROCESS)

SHEET TWO

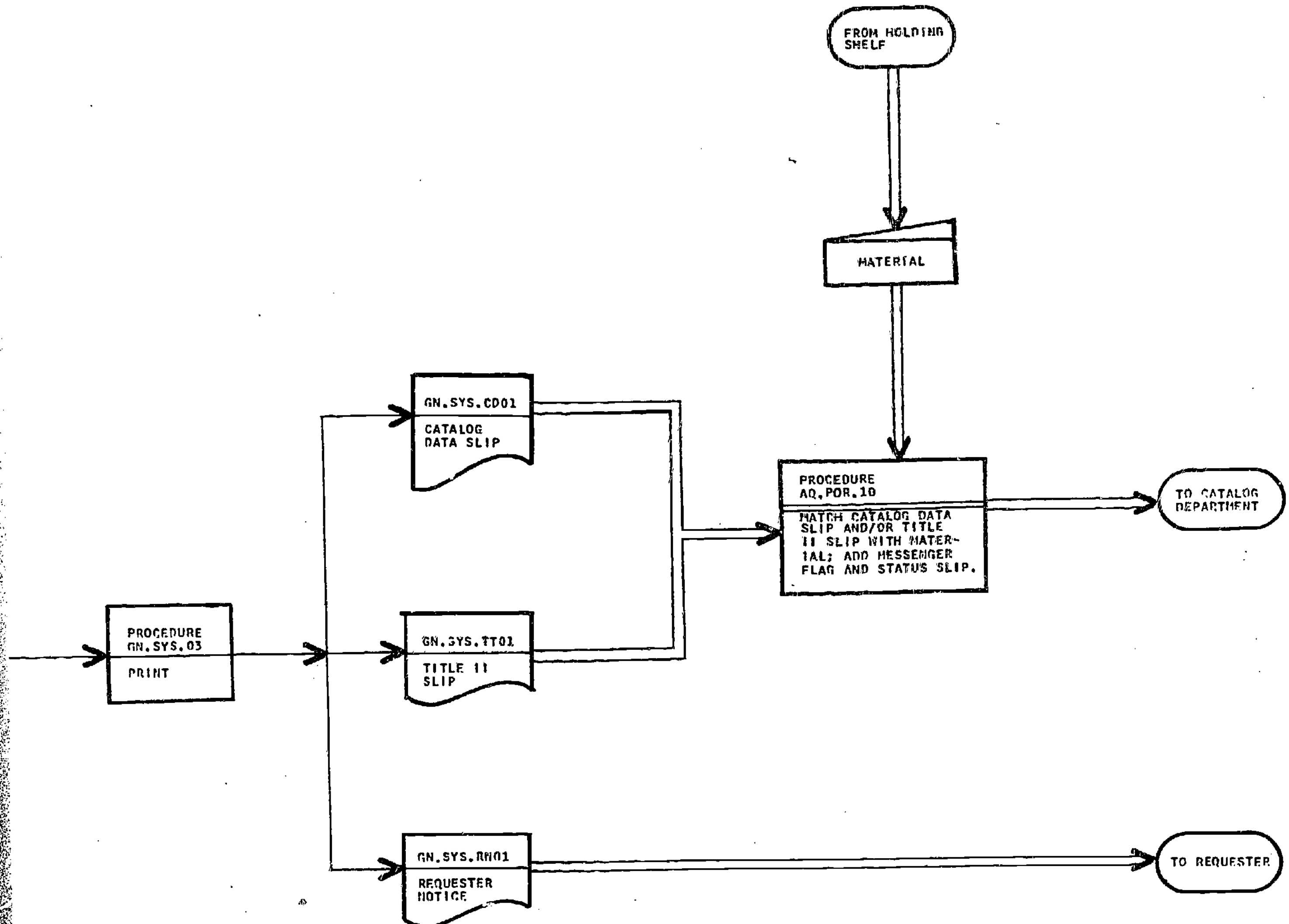


BIPF
IPF
GN.SYS.OF02
OVERFLOW
GN.SYS.BU01-06
UPDATE -BIBLIO-GRAPHIC (WITH ERROR MESSAGES)
DATA ELEMENT ERROR
GN.SYS.OF02
UPDATE - OVERFLOW
PROCEDURE AQ.POR.03
KEY BIBLIOGRAPHIC UPDATE
GN.SYS.BU01-06
UPDATE - BIBLIOGRAPHIC
PROCEDURE GN.SYS.01
UPDATE IPF
ACCEPTABLE BIBLIOGRAPHIC VARIATION
PROCEDURE AQ.POR.02
MATCH MATERIAL WITH DEALER REPORT
PROCEDURE AQ.POR.04
KEY PURCHASE ORDER MRI
IDENTICAL BIBLIOGRAPHIC MATCH
PROCEDURE AQ.POR.05
WRITE IDX IN MATERIAL
MATERIAL
UNACCEPTABLE BIBLIOGRAPHIC VARIATION; TO BE RETURNED AND CORRECT ITEM CLAIMED
TO AQ.CAN
TO AQ.MCL
TO HOLDING SHELF



SHEET THREE
POR/01
2.4
GN.SYS.OF02
GN.SYS.MR01
UPDATE PURCHASE ORDER MATERIAL RECEIPT (WITH ERROR MESSAGE)
DATA ELEMENT ERROR
GN.SYS.OF02
UPDATE OVERFLOW
GN.SYS.MR01
UPDATE PURCHASE ORDER MATERIAL RECEIPT
PROCEDURE GN.SYS.01
UPDATE IPF
GN.SYS.PT05
CATALOG DATA SLIP PRINT RECORDS
GN.SYS.PT06
TITLE II SLIP PRINT RECORDS
GN.SYS.PT07
REQUESTER NOTICE PRINT RECORDS
BIPF
IPF

SHEET FOUR



STANFORD UNIVERSITY LIBRARIES

APPROVAL(s) REQUIRED: RALPH HANSEN, FRED LYNDEN

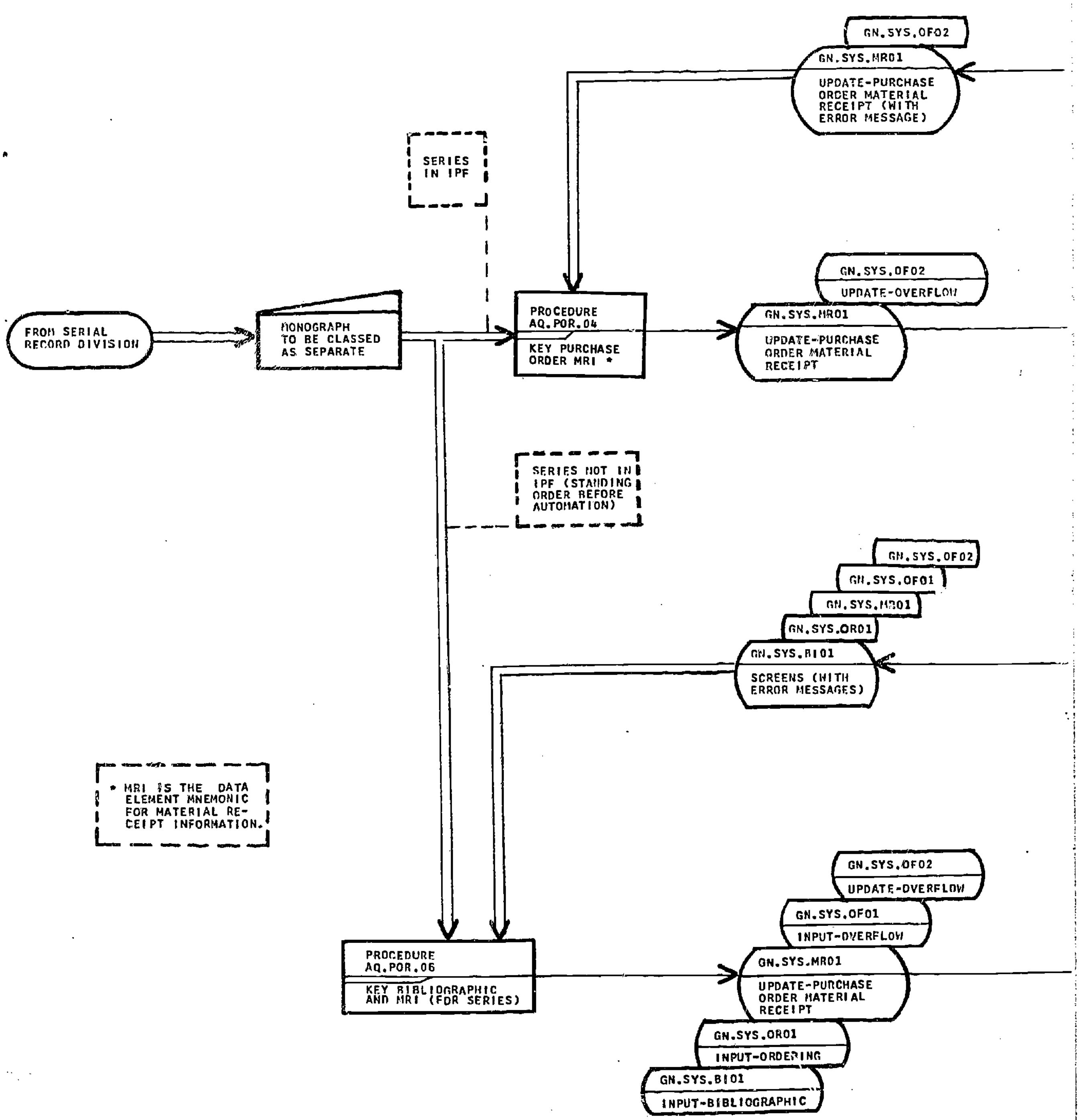
REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEP INI
0	10/20	INITIAL VERSION	PD		

TITLE: PURCHASE ORDER MATERIAL RECEIPT PROCESS

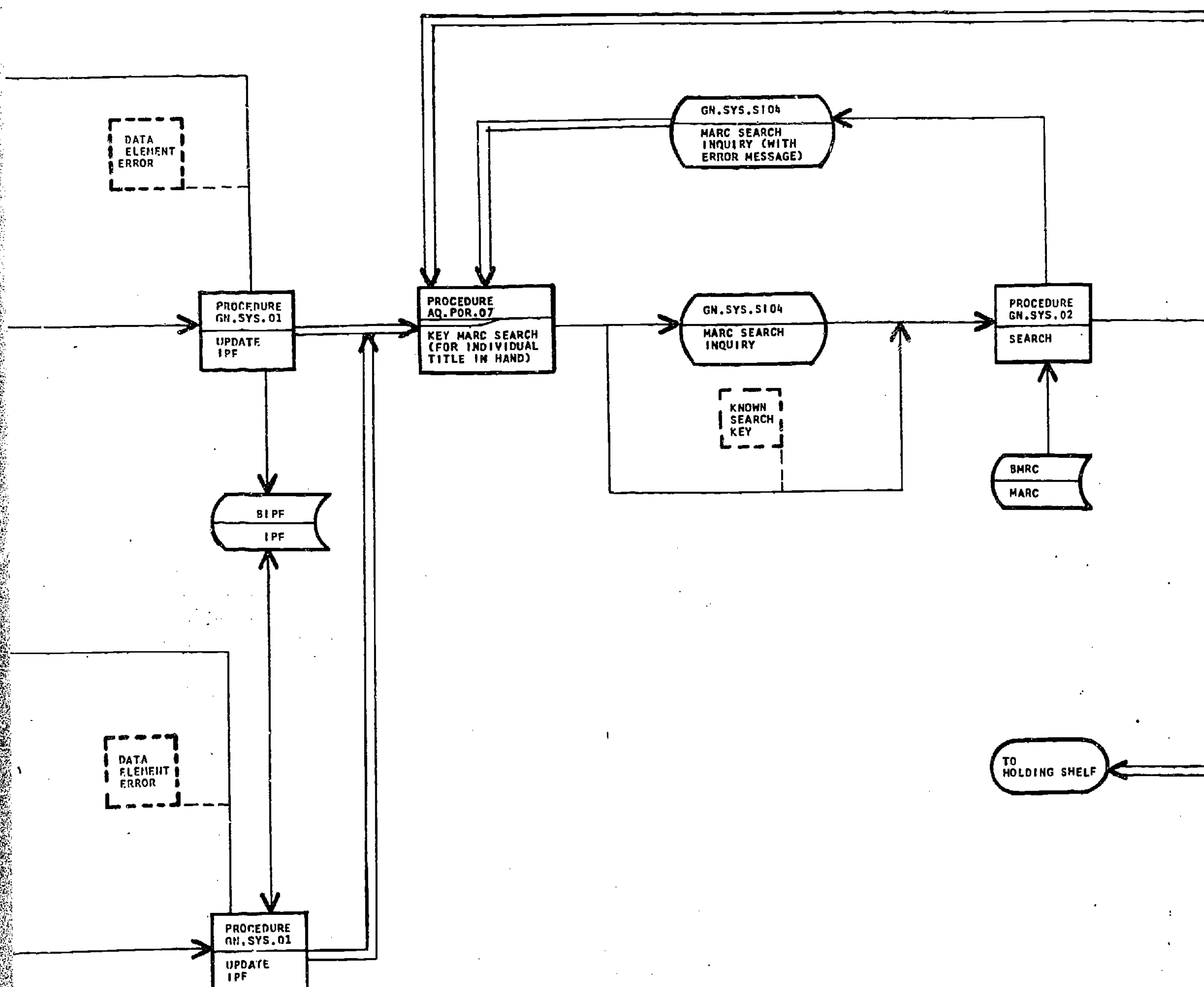
ID: AQ.POR

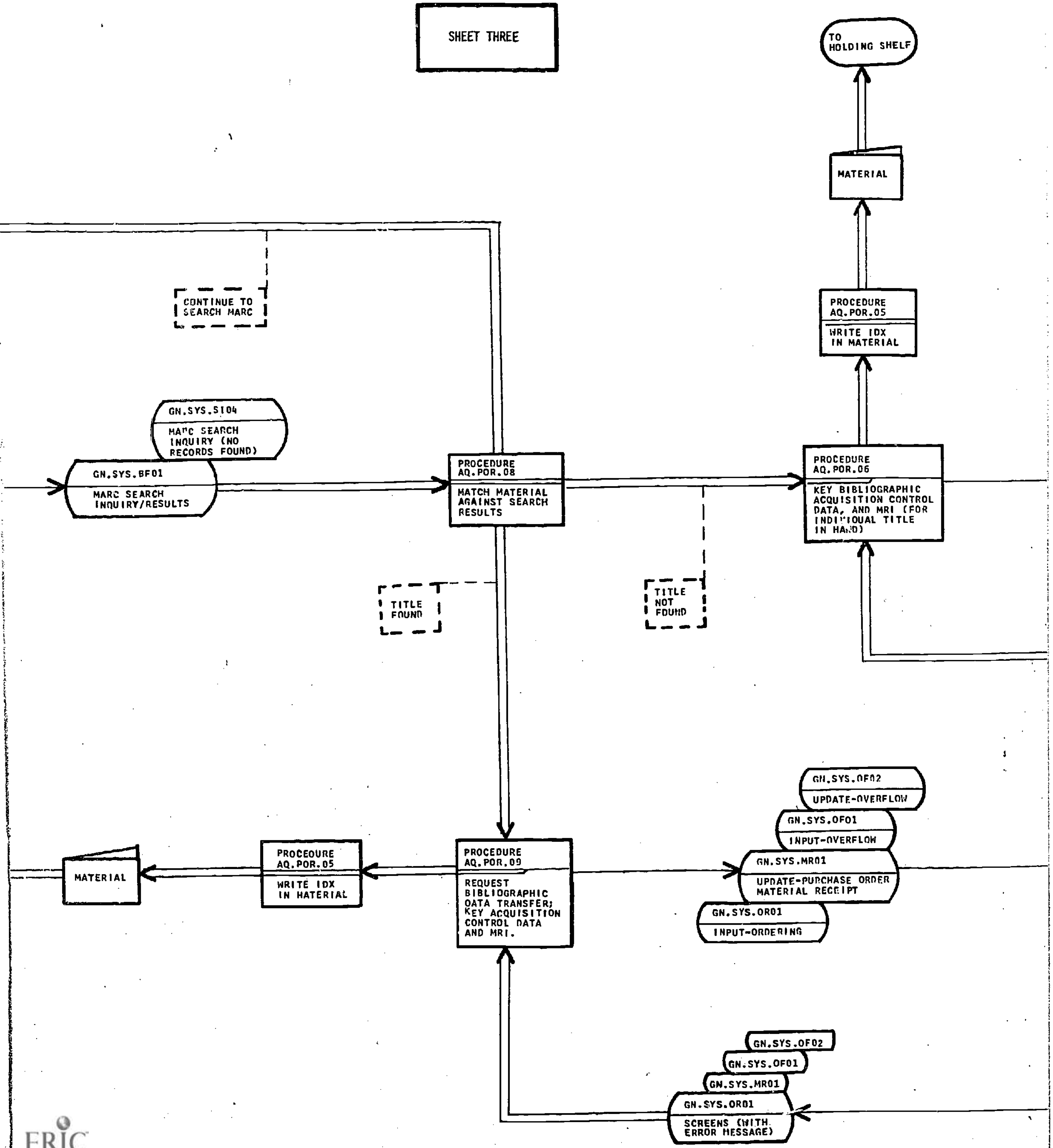
4 SHEETS PAGE 1 OF 2

SHEET ONE

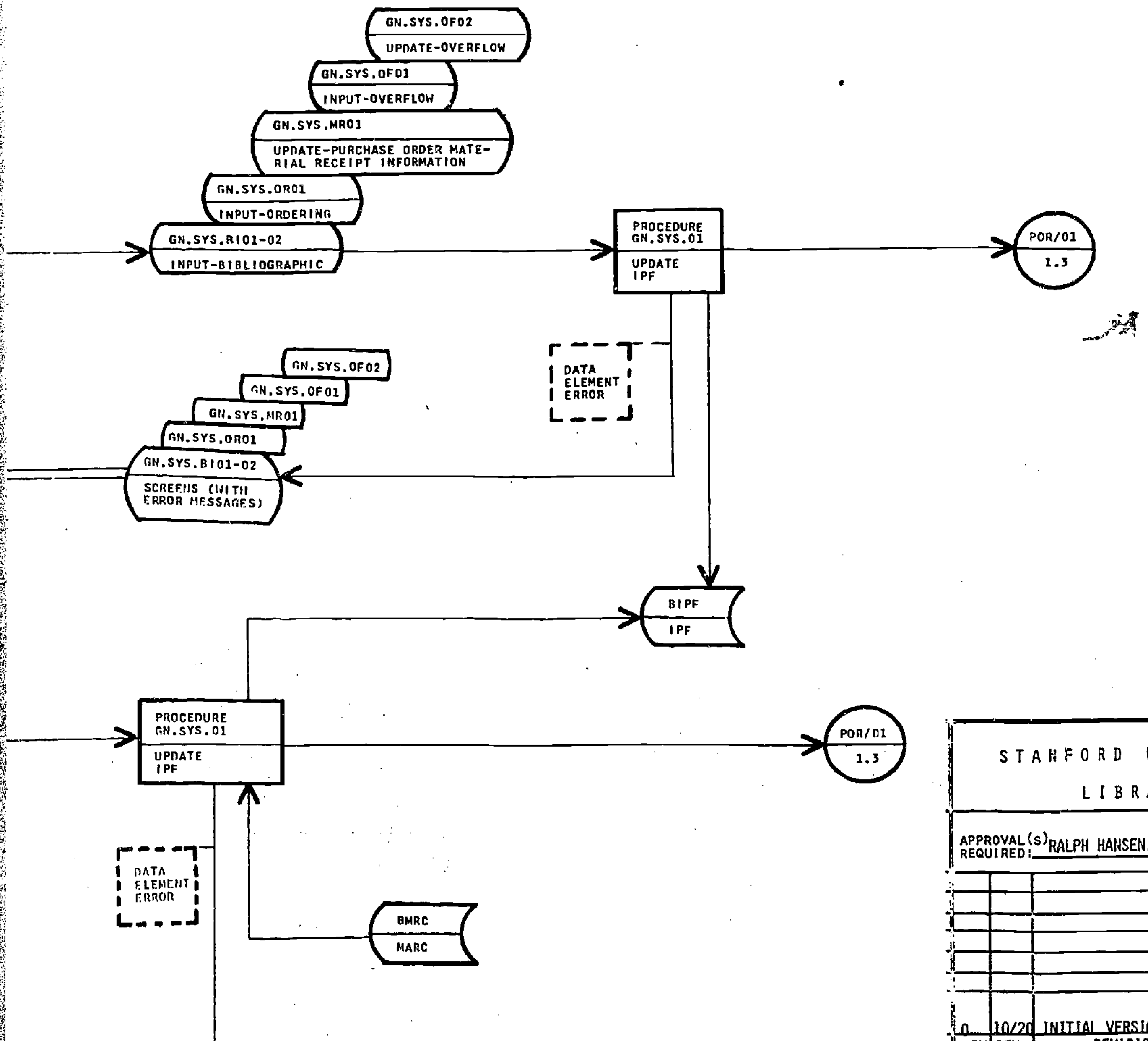

GN.SYS.OF02
GN.SYS.MR01
UPDATE-PURCHASE ORDER MATERIAL RECEIPT (WITH ERROR MESSAGE)
SERIES IN IPF
FROM SERIAL RECORD DIVISION
MONOGRAPH TO BE CLASSED AS SEPARATE
PROCEDURE AQ.POR.04
KEY PURCHASE ORDER MRI *
GN.SYS.OF02
UPDATE-OVERFLOW
GN.SYS.MR01
UPDATE-PURCHASE ORDER MATERIAL RECEIPT
SERIES NOT IN IPF (STANDING ORDER BEFORE AUTOMATION)
GN.SYS.OF02
GN.SYS.OF01
GN.SYS.MR01
GN.SYS.OR01
GN.SYS.BI01
SCREENS (WITH ERROR MESSAGES)
* MRI IS THE DATA ELEMENT MNEMONIC FOR MATERIAL RECEIPT INFORMATION.
PROCEDURE AQ.POR.06
KEY BIBLIOGRAPHIC AND MRI (FOR SERIES)
GN.SYS.OF02
UPDATE-OVERFLOW
GN.SYS.OF01
INPUT-OVERFLOW
GN.SYS.MR01
UPDATE-PURCHASE ORDER MATERIAL RECEIPT
GN.SYS.OR01
INPUT-ORDERING
GN.SYS.BI01
INPUT-BIBLIOGRAPHIC

SHEET TWO


DATA ELEMENT ERROR
PROCEDURE GN.SYS.01
UPDATE IPF
BIPF
IPF
DATA ELEMENT ERROR
PROCEDURE GN.SYS.01
UPDATE IPF
PROCEDURE AQ.POR.07
KEY MARC SEARCH (FOR INDIVIDUAL TITLE IN HAND)
GN.SYS.S104
MARC SEARCH INQUIRY (WITH ERROR MESSAGE)
GN.SYS.S104
MARC SEARCH INQUIRY
KNOWN SEARCH KEY
PROCEDURE GN.SYS.02
SEARCH
BMRC
MARC
TO HOLDING SHELF


SHEET THREE
TO HOLDING SHELF
MATERIAL
PROCEDURE AQ.POR.05
WRITE IDX IN MATERIAL
CONTINUE TO SEARCH MARC
GN.SYS.SI04
MARC SEARCH INQUIRY (NO RECORDS FOUND)
GN.SYS.BF01
MARC SEARCH INQUIRY/RESULTS
PROCEDURE AQ.POR.08
MATCH MATERIAL AGAINST SEARCH RESULTS
PROCEDURE AQ.POR.06
KEY BIBLIOGRAPHIC ACQUISITION CONTROL DATA, AND MRI (FOR INDIVIDUAL TITLE IN HAND)
TITLE FOUND
TITLE NOT FOUND
GN.SYS.OF02
UPDATE-OVERFLOW
GN.SYS.OF01
INPUT-OVERFLOW
GN.SYS.MR01
UPDATE-PURCHASE ORDER MATERIAL RECEIPT
GN.SYS.OR01
INPUT-ORDERING
MATERIAL
PROCEDURE AQ.POR.05
WRITE IDX IN MATERIAL
PROCEDURE AQ.POR.09
REQUEST BIBLIOGRAPHIC DATA TRANSFER; KEY ACQUISITION CONTROL DATA AND MRI.
GN.SYS.OF02
GN.SYS.OF01
GN.SYS.MR01
GN.SYS.OR01
SCREENS (WITH ERROR MESSAGE)

SHEET FOUR



STANFORD UNIVERSITY LIBRARIES

APPROVAL(S) REQUIRED: RALPH HANSEN, FRED LYNDEN

REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	D I
0	10/20	INITIAL VERSION	PD		

TITLE: PURCHASE ORDER MATERIAL RECEIPT PROCESS

ID: AQ.POR

4 SHEETS PAGE 2 OF 2

DISTRIBUTION AND MARC SEARCHING PROCESS (CT.DIS)

The central distribution section within the Catalog Department receives from the Acquisition Department material ordered through the automated Acquisition Subsystem, along with a Catalog Data Slip (GN.SYS.CD01), a Catalog Data Slip Copy (GN.SYS.CD02), and a Title II Slip (GN.SYS.TT01) if applicable. It receives material ordered through the manual system along with one to three copies of an SUL-7 form (CT.DIS.SU01), and perhaps a book requisition (AQ.SEL.BR01) and/or a Title II card (AQ.ORD.LC01). Messenger slips and slips indicating added copy, added volume, and serial analytic come with the material as needed. Serial analytics are accompanied by a Charge Slip (CT.DIS.CS01).

Material will be distributed as follows:

1. Material requiring original cataloging: if the material is for stack, reference room, graduate program in humanities, Catalog Department collection, or modern European languages, the IPF record is updated to show the first one or two letters of a tentative LC classification (used for counting arrears and for locating uncataloged material), and the material then is sent to the cataloger for Original Cataloging (CT.ORG). If the material is for other locations, it is sent directly to the cataloger (CT.ORG).

2. Material for which LC data is expected: this is placed in the holding area in ID number sequence, and the Title II slip filed.

3. Material for Meyer Library: this is placed in the holding area in ID number sequence if LC data is expected, and held up to two months. Otherwise it is sent to Meyer Cataloging (CT.MEY).

4. Overseas campuses material: this is sent to Overseas Cataloging (CT.OVS).

5. Added copies, added volumes, and replacement copies: these are sent to Added Copies/Volumes (CT.ACV).

6. Material for which LC data has been found: this is sent to LC Cataloging (CT.LCC).

7. Purchase order material for which MARC data is expected: the distributor searches the MARC file. If he does not find MARC data, he creates a MARC standing search request and puts the material in the holding area; if he finds MARC data, he transfers it to the IPF, and a MARC Data Slip (GN.SYS.MD01-02) is created and matched with the material, which is then sent to CT.LCC.

8. Material that needs an NPAC notice, a standing search request for MARC data, or a Title II Slip: the IPF record is updated to produce the required output; and the material is placed in the holding area.

The distribution section receives a weekly Matched Records Listing of the ID numbers of titles for which MARC data has been found on a MARC tape during MARC Processing (TP.MAR). The distributor verifies the match by calling up the IPF record and the corresponding MARC record on a CRT screen (GN.SYS.BF01) and, for successful matches, initiates the transfer of data from the MARC File to the In Process File. The IPF record is updated automatically as part of the transfer operation to show "MARC data found." As a result of the transfer, a MARC Data Slip (GN.SYS.MD01) and a MARC Data Slip Copy (GN.SYS.MD02) are produced and matched with the material, which then is sent to LC Cataloging or Meyer Cataloging.

If a Title II card (AQ.ORD.LC01) is found while filing is being done in the Title II File, the card and the Title II Slip (GN.SYS.TT01) are pulled and matched with the material being held in the holding area. (If the material is for Meyer, the Title II card is not pulled; the Title II Slip is annotated to show the presence of a Title II card and placed with the material.) The IPF record is updated to show "Title II card found" and the material is sent to LC Cataloging or Meyer Cataloging.

The distribution section receives a monthly Out of Date Standing Search Requests Listing (GN.SYS.OL01) of the items for which MARC or Title II data has not been found during the designated holding period. The material is pulled from the holding area and previous searching is updated as necessary (including a final on-line search of the MARC file). The IPF record is updated with the tentative class number, if necessary, and the material is sent to Original Cataloging or Meyer Cataloging, or to LC Cataloging if LC data was found while searching was being updated.

The Out of Date Standing Search Requests Listing does not include material ordered under the manual system. Such material is shelved in the holding area by date of receipt and pulled at the end of the holding period as in the current manual system.

Bulk acquisition material is received by the central distribution section from its source; it does not go through the Acquisition Department, and has no IPF record. This material is manually controlled and distributed for Original Cataloging. This manual procedure is not shown on the CT.DIS flow chart.

When the distribution section receives Catalog Requests (GN.SYS.RC01 and GN.SYS.RC02) and Requests for Book in Process (CT.DIS.RB01), the distributor must determine the location of the material within the Catalog Department. (If the request does not give enough information to locate the item, an IPF search may be made.) If the material has been distributed for cataloging, he notifies the cataloger that the request has been received and that the material should be cataloged on a priority basis. If the material is in the holding area, he pulls it and pulls the Title II slip if necessary. He updates the IPF record with the temporary classification and gives the material to the cataloger (CT.ORG or CT.MEY).

Material that is not received by the central distribution section and thus does not follow the flow outlined above will be handled in a similar fashion by the appropriate units as described below.

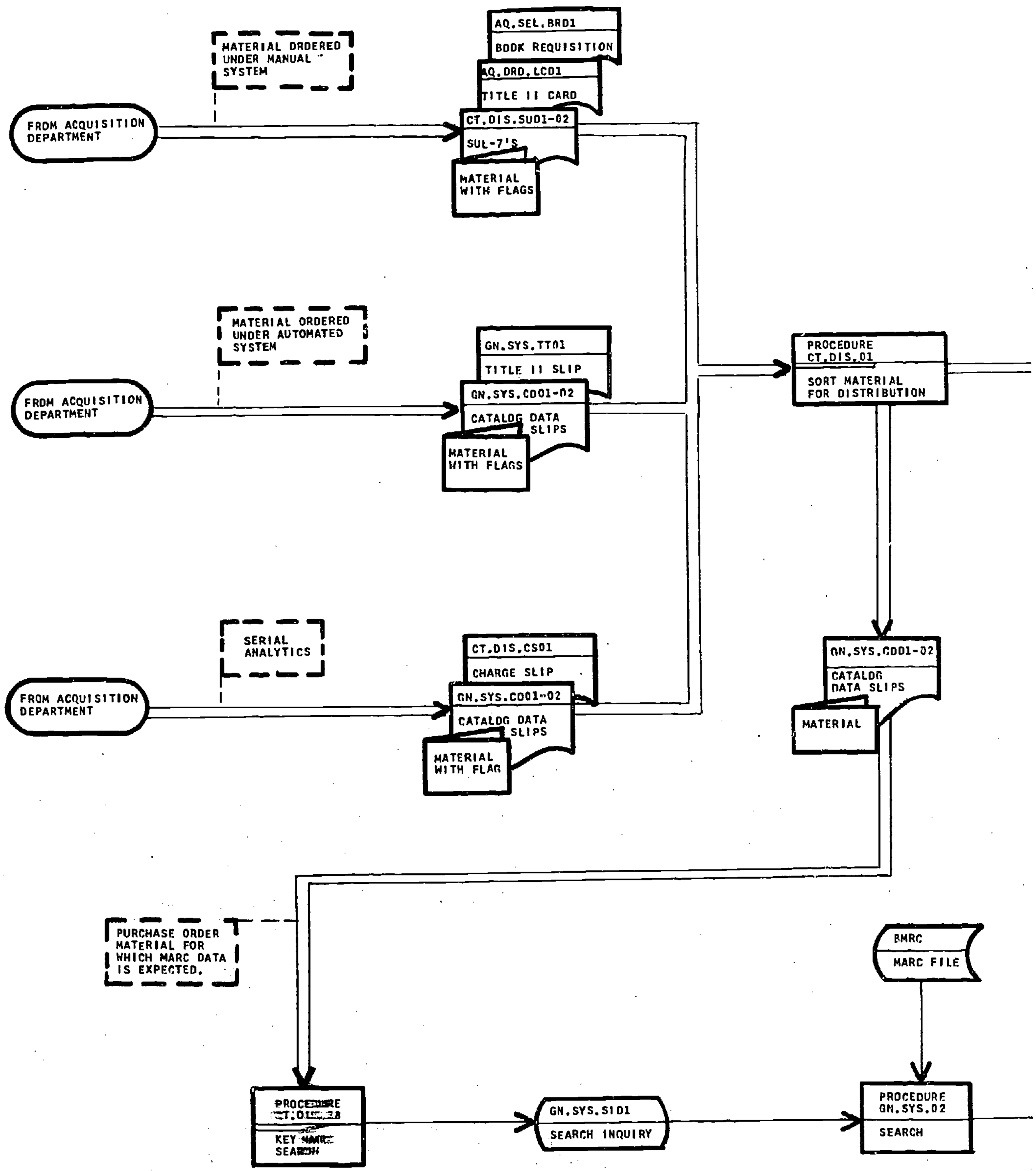
1. Music materials ordered through the Automated system: these are sent with Catalog Data Slips and Title II Slips by the Acquisition Department directly to the Music Library, where they are distributed within the Music Cataloging Section to Original Cataloging, LC Cataloging, or Added Copies/Volumes, or placed in a holding area to await LC data. The Title II Slips are filed in the LC Card File in the music library. If a Title II Card is found there, it is pulled and placed with the material that is sent to CT.LCC. In this case the IPF record is not updated to show that LC data has been found. If MARC data is found for Music material, the transfer of data to the IPF is done at the Main Library and the distribution section send the MARC Data Slips to the music Library for matching with the material, which then is sent to CT.LCC. If MARC data is not found during the holding period, the distribution section sends the monthly Out of Date Standing Search Requests Listing for Music items to the Music Library so that the Music Cataloger may pull the material from his holding area and distribute it for Original Cataloging.

2. Special collections materials: these are sent directly from the Acquisition Department to Special Collections, where they are distributed for Original Cataloging or LC Cataloging.

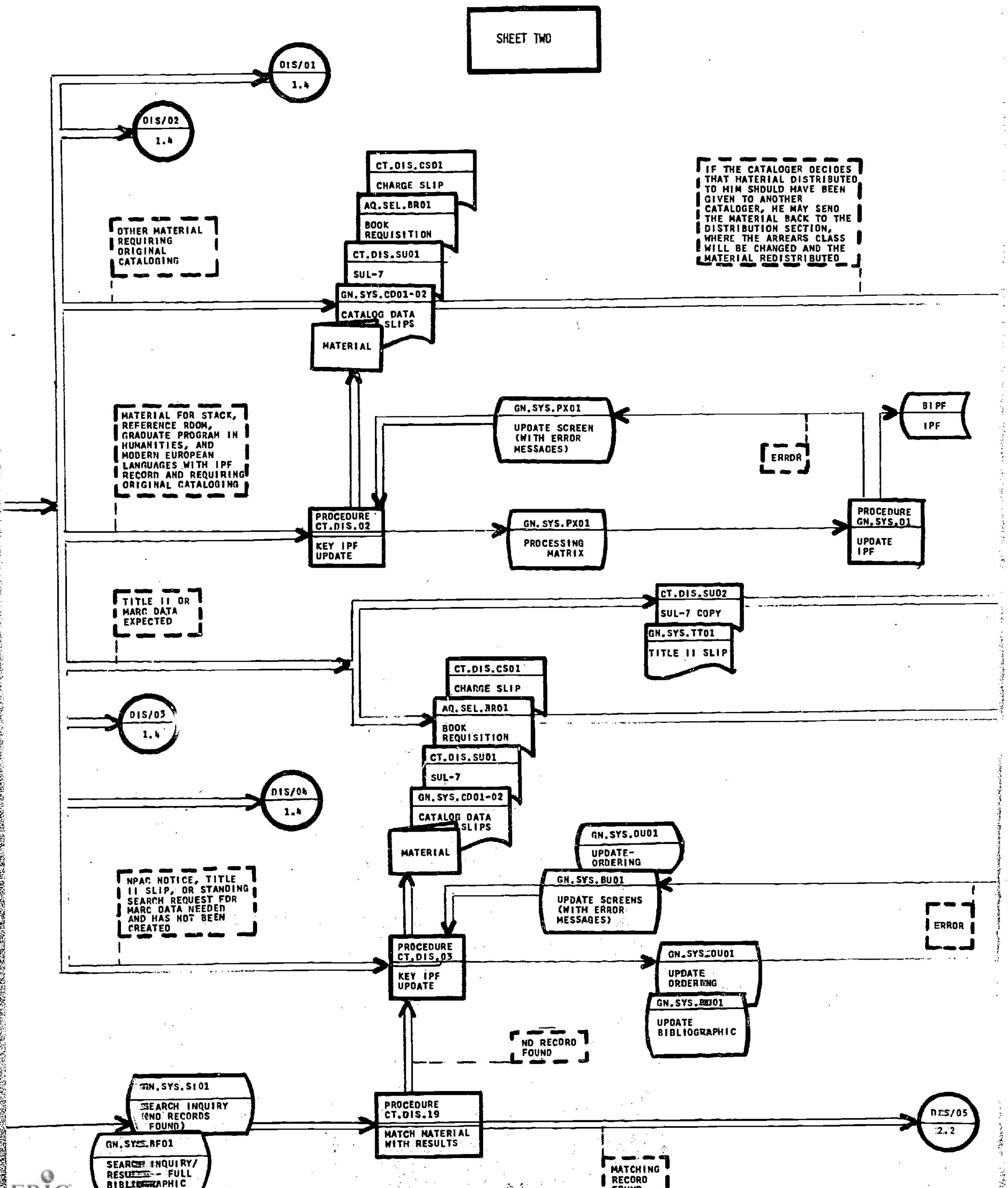
3. PL480 program materials from Yugoslavia (no IPF record): these are received in the Slavic unit with a PL480 Mimeographed Card (CT.DIS.M001). The card is filed in the Title II file and the material is held in the Slavic Cataloging unit. If a Title II card is found, the distribution section sends it the the Slavic unit. No automatic notification of the end of the holding period is provided.

4. Serial titles new to Stanford University Libraries: these titles are received in Serial Records, where they are matched with the Process Slips (GN.SYS.PS01) produced at the time of ordering. They are then sent to the Serials Cataloging unit, where they are distributed. Material requiring Original Cataloging is sent to the appropriate cataloger (CT.ORG); material with LC data is cataloged in the Serials unit (CT.LCC). If Title II data is expected, the material is kept in a holding area and searched repetitively in the Title II file.

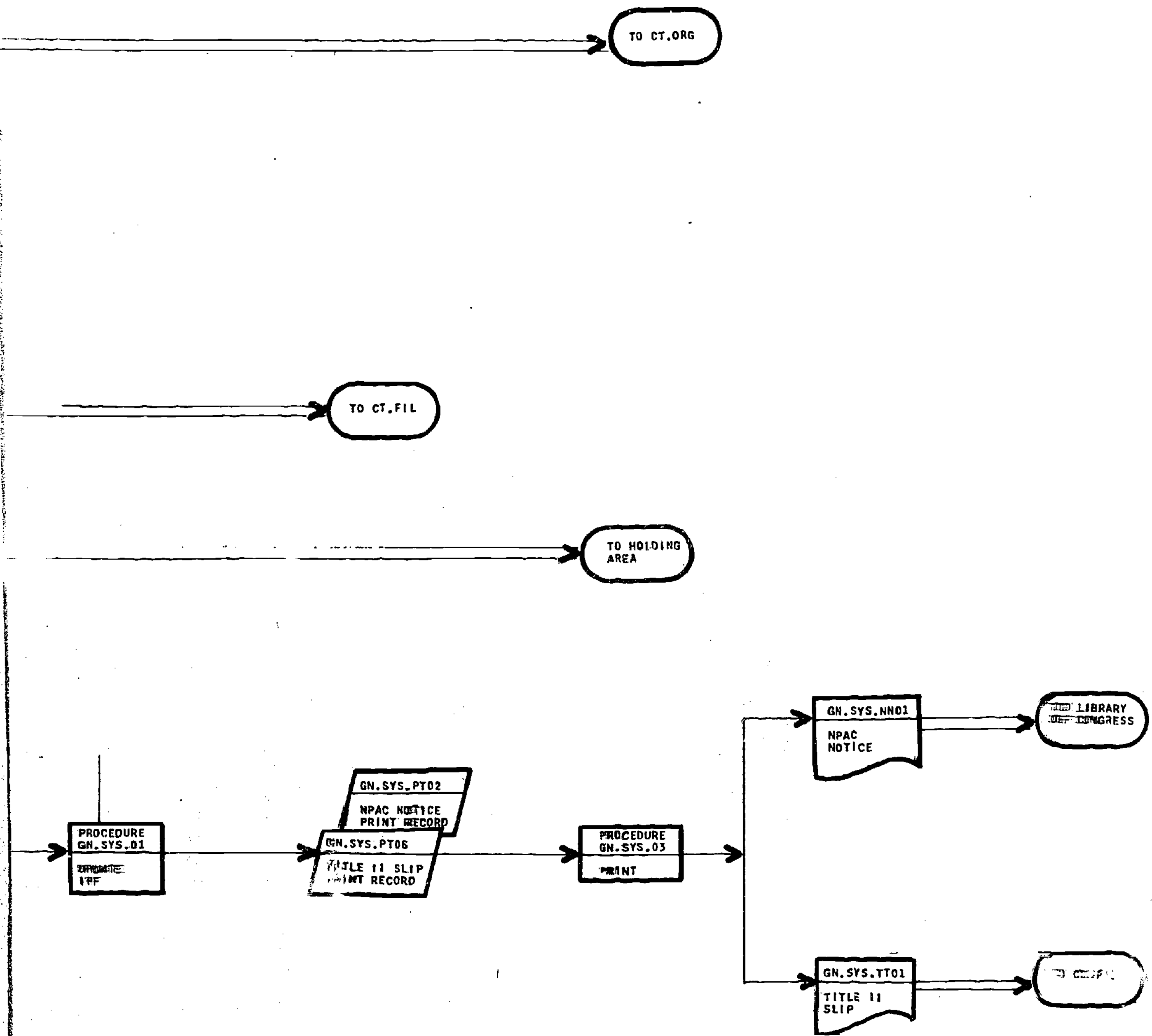
SHEET ONE



FROM ACQUISITION DEPARTMENT
MATERIAL ORDERED UNDER MANUAL SYSTEM
AQ.SEL.BRD1
BOOK REQUISITION
AQ.ORD.LCO1
TITLE II CARD
CT.DIS.SUD1-02
SUL-7'S
MATERIAL WITH FLAGS
FROM ACQUISITION DEPARTMENT
MATERIAL ORDERED UNDER AUTOMATED SYSTEM
GN.SYS.TT01
TITLE II SLIP
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL WITH FLAGS
PROCEDURE CT.DIS.01
SORT MATERIAL FOR DISTRIBUTION
FROM ACQUISITION DEPARTMENT
SERIAL ANALYTICS
CT.DIS.CS01
CHARGE SLIP
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL WITH FLAG
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
PURCHASE ORDER MATERIAL FOR WHICH MARC DATA IS EXPECTED.
BMRC
MARC FILE
PROCEDURE
KEY
SEARCH
GN.SYS.SI01
SEARCH INQUIRY
PROCEDURE GN.SYS.02
SEARCH

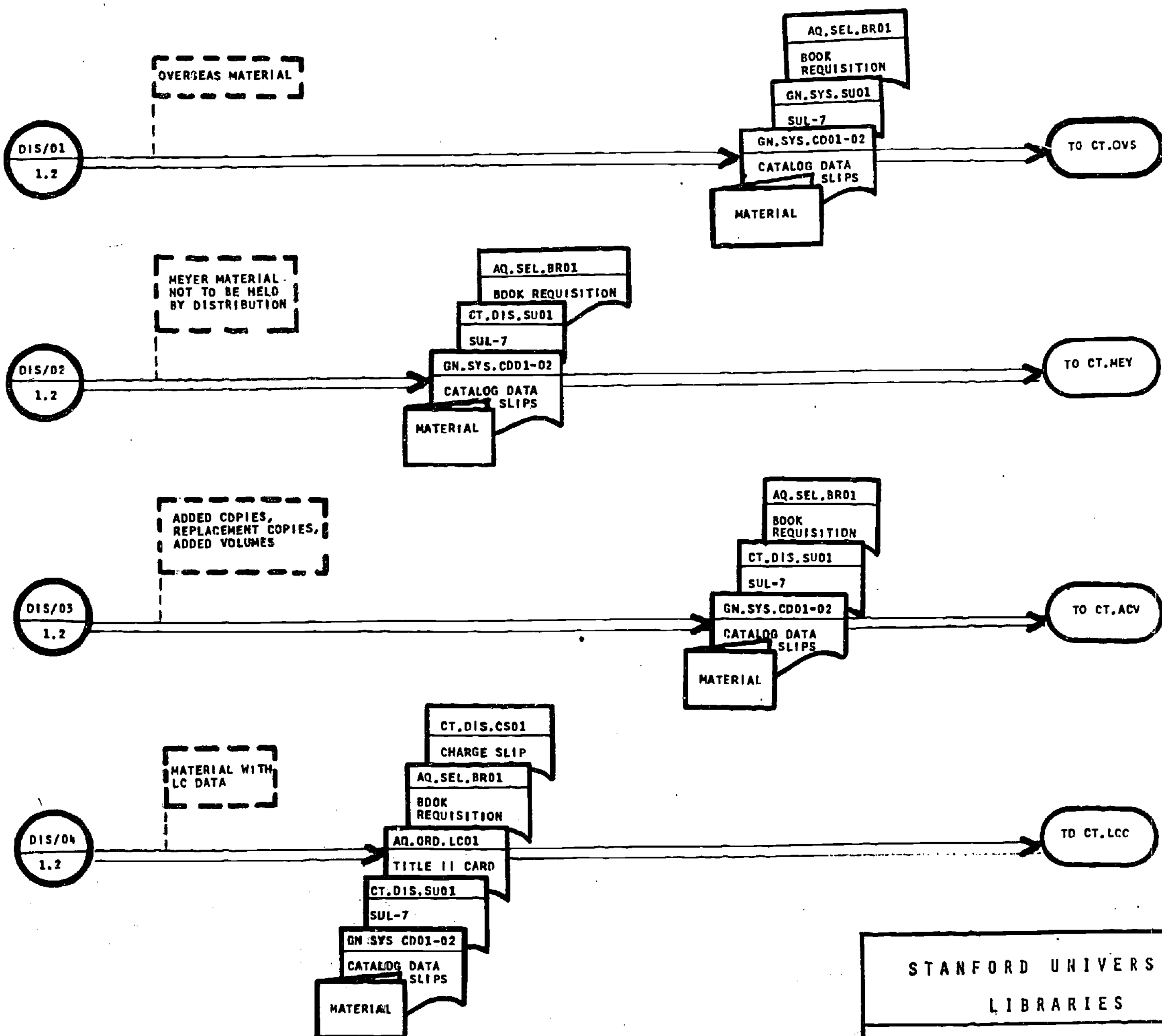

SHEET TWO
DIS/01
1.4
DIS/02
1.4
CT.DIS.CS01
CHARGE SLIP
AQ.SEL.BR01
BOOK REQUISITION
CT.DIS.SU01
SUL-7
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
OTHER MATERIAL REQUIRING ORIGINAL CATALOGING
IF THE CATALOGER DECIDES THAT MATERIAL DISTRIBUTED TO HIM SHOULD HAVE BEEN GIVEN TO ANOTHER CATALOGER, HE MAY SEND THE MATERIAL BACK TO THE DISTRIBUTION SECTION, WHERE THE ARREARS CLASS WILL BE CHANGED AND THE MATERIAL REDISTRIBUTED
MATERIAL FOR STACK, REFERENCE ROOM, GRADUATE PROGRAM IN HUMANITIES, AND MODERN EUROPEAN LANGUAGES WITH IPF RECORD AND REQUIRING ORIGINAL CATALOGING
GN.SYS.PX01
UPDATE SCREEN (WITH ERROR MESSAGES)
ERROR
BIPF
IPF
PROCEDURE CT.DIS.02
KEY IPF UPDATE
GN.SYS.PX01
PROCESSING MATRIX
PROCEDURE GN.SYS.01
UPDATE IPF
TITLE II OR MARC DATA EXPECTED
CT.DIS.SU02
SUL-7 COPY
GN.SYS.TT01
TITLE II SLIP
CT.DIS.CS01
CHARGE SLIP
AQ.SEL.BR01
BOOK REQUISITION
CT.DIS.SU01
SUL-7
GN.SYS.CD01-02
CATALOG DATA SLIPS
DIS/03
1.4
DIS/04
1.4
MATERIAL
GN.SYS.OU01
UPDATE-ORDERING
GN.SYS.BU01
UPDATE SCREENS (WITH ERROR MESSAGES)
NPAC NOTICE, TITLE II SLIP, OR STANDING SEARCH REQUEST FOR MARC DATA NEEDED AND HAS NOT BEEN CREATED
ERROR
PROCEDURE CT.DIS.03
KEY IPF UPDATE
UPDATE ORDERING
UPDATE BIBLIOGRAPHIC
NO RECORD FOUND
SEARCH INQUIRY (NO RECORDS FOUND)
PROCEDURE CT.DIS.19
MATCH MATERIAL WITH RESULTS
DIS/05
2.2
MATCHING RECORD FOUND

SHEET THREE



TO CT.ORG
TO CT.FIL
TO HOLDING AREA
PROCEDURE GN.SYS.01
GN.SYS.PT02
NPAC NOTICE PRINT RECORD
GN.SYS.PT06
TITLE II SLIP
PROCEDURE GN.SYS.03
PRINT
GN.SYS.NN01
NPAC NOTICE
LIBRARY
GN.SYS.TT01
TITLE II SLIP

SHEET FOUR



STANFORD UNIVERSITY LIBRARIES

APPROVAL(s) REQUIRED: J. HITCHCOCK

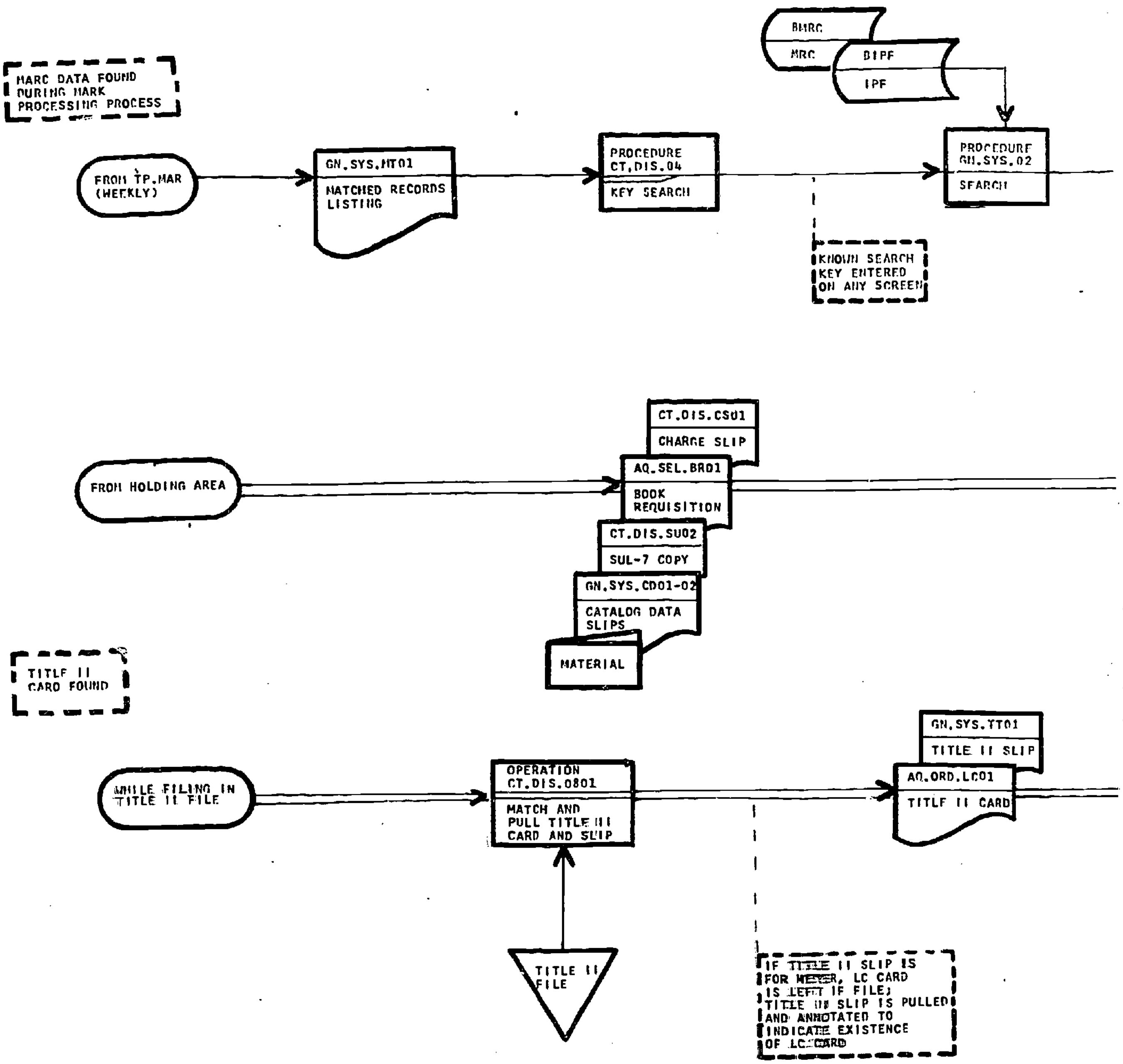
REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEPT INIT
0	1/22	INITIAL VERSION	LK		

TITLE: DISTRIBUTION AND MARC SEARCHING PROCESS

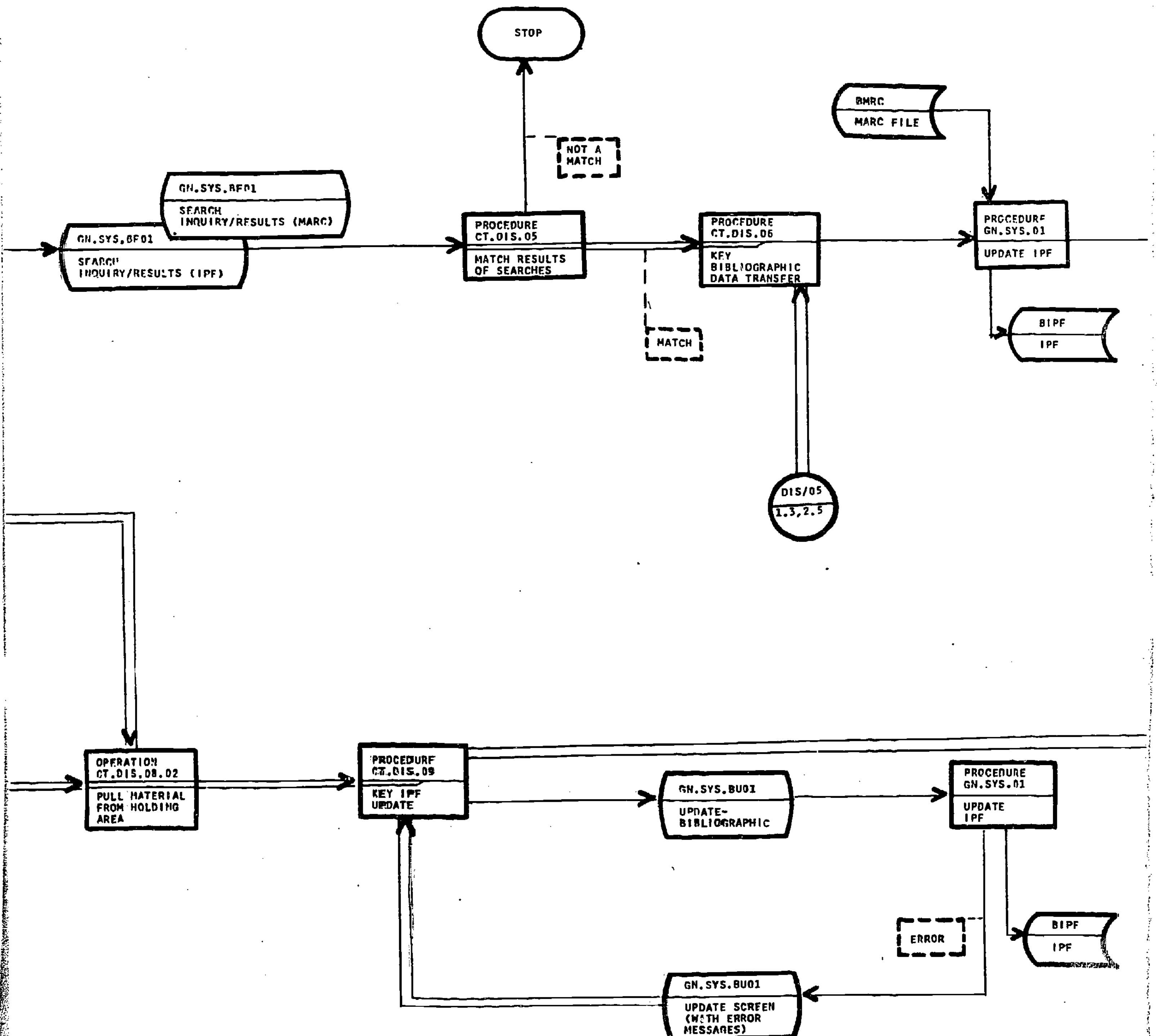
ID: CT.DIS

4 SHEETS PAGE 1 OF 5

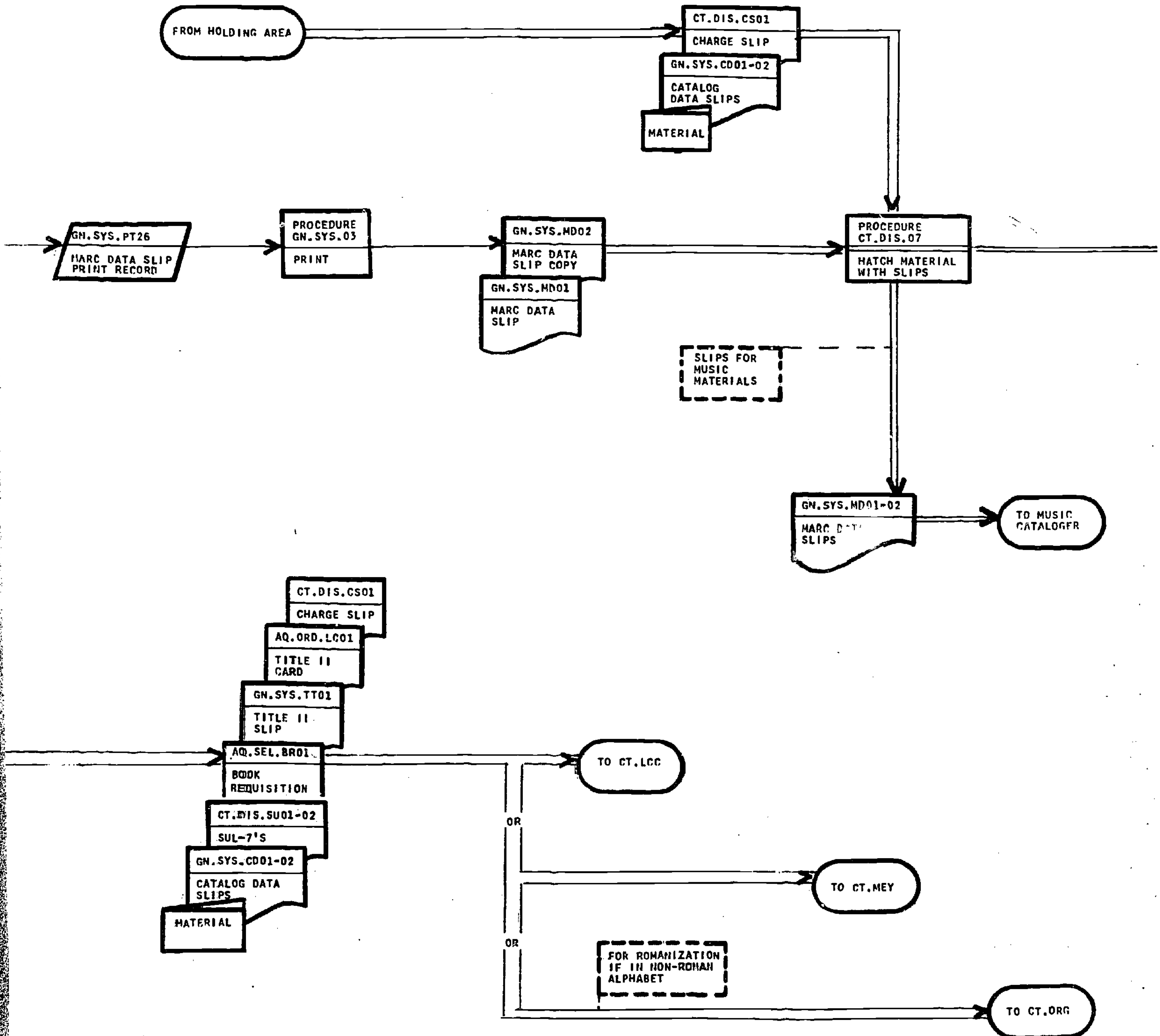
SHEET ONE


BMRC
MRC
BIPF
IPF
MARC DATA FOUND DURING MARK PROCESSING PROCESS
FROM TP.MAR (WEEKLY)
GN.SYS.MT01
MATCHED RECORDS LISTING
PROCEDURE CT.DIS.04
KEY SEARCH
PROCEDURE GN.SYS.02
SEARCH
KNOWN SEARCH KEY ENTERED ON ANY SCREEN
CT.DIS.CS01
CHARGE SLIP
FROM HOLDING AREA
AQ.SEL.BR01
BOOK REQUISITION
CT.DIS.SU02
SUL-7 COPY
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
TITLE II CARD FOUND
GN.SYS.TT01
TITLE II SLIP
WHILE FILING IN TITLE II FILE
OPERATION CT.DIS.0801
MATCH AND PULL TITLE II CARD AND SLIP
AQ.ORD.LC01
TITLE II CARD
TITLE II FILE
IF TITLE II SLIP IS FOR MEYER, LC CARD IS LEFT IF FILE; TITLE II SLIP IS PULLED AND ANNOTATED TO INDICATE EXISTENCE OF LC CARD

SHEET TWO

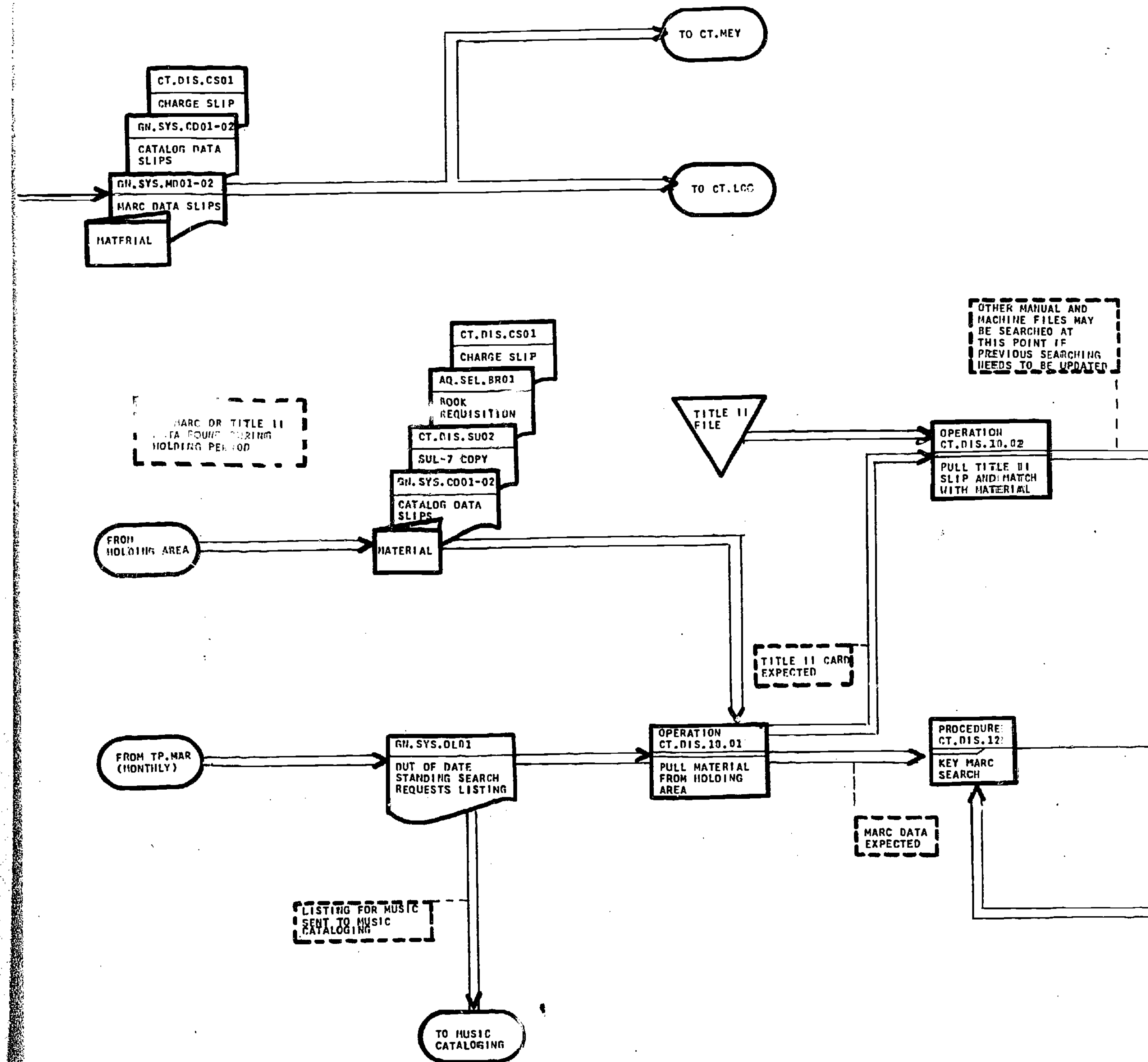

STOP
NOT A MATCH
BMRC
MARC FILE
GN.SYS.BF01
SEARCH INQUIRY/RESULTS (MARC)
GN.SYS.BF01
SEARCH INQUIRY/RESULTS (IPF)
PROCEDURE CT.DIS.05
MATCH RESULTS OF SEARCHES
PROCEDURE CT.DIS.06
KEY BIBLIOGRAPHIC DATA TRANSFER
PROCEDURE GN.SYS.01
UPDATE IPF
MATCH
BIPF
IPF
DIS/05
1.3,2.5
OPERATION CT.DIS.08.02
PULL MATERIAL FROM HOLDING AREA
PROCEDURE CT.DIS.09
KEY IPF UPDATE
GN.SYS.BU01
UPDATE-BIBLIOGRAPHIC
PROCEDURE GN.SYS.01
UPDATE IPF
ERROR
BIPF
IPF
GN.SYS.BU01
UPDATE SCREEN (WITH ERROR MESSAGES)

SHEET THREE



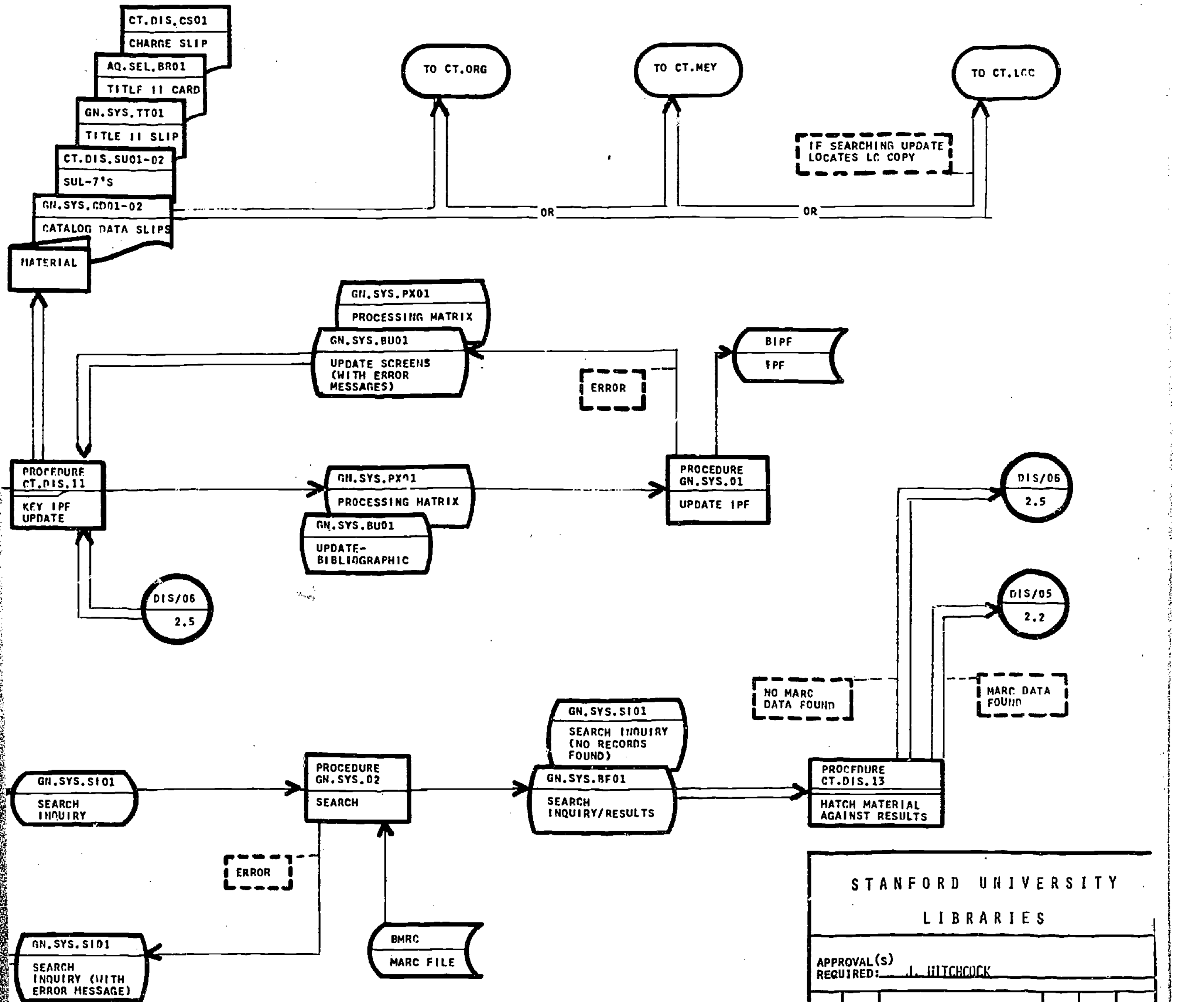
FROM HOLDING AREA
CT.DIS.CS01
CHARGE SLIP
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
GN.SYS.PT26
MARC DATA SLIP PRINT RECORD
PROCEDURE GN.SYS.03
PRINT
GN.SYS.MD02
MARC DATA SLIP COPY
GN.SYS.MD01
MARC DATA SLIP
PROCEDURE CT.DIS.07
MATCH MATERIAL WITH SLIPS
SLIPS FOR MUSIC MATERIALS
GN.SYS.MD01-02
SLIPS
TO MUSIC CATALOGER
CT.DIS.CS01
CHARGE SLIP
AQ.ORD.LC01
TITLE II CARD
GN.SYS.TT01
TITLE II SLIP
AQ.SEL.BR01
BOOK REQUISITION
SUL-7'S
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
TO CT.LCC
OR
TO CT.MEY
OR
FOR ROMANIZATION IF IN NON-ROMAN ALPHABET
TO CT.ORG

SHEET FOUR



TO CT.MEY
CT.DIS.CS01
CHARGE SLIP
GN.SYS.CD01-02
CATALOG DATA SLIPS
GN.SYS.MD01-02
MARC DATA SLIPS
MATERIAL
TO CT.LCC
OTHER MANUAL AND MACHINE FILES MAY BE SEARCHED AT THIS POINT IF PREVIOUS SEARCHING NEEDS TO BE UPDATED
CT.DIS.CS01
CHARGE SLIP
AQ.SEL.BR01
BOOK REQUISITION
MARC OR TITLE II
CT.DIS.SU02
SUL-7 COPY
GN.SYS.CD01-02
CATALOG DATA SLIPS
TITLE II FILE
OPERATION CT.DIS.10.02
PULL TITLE II SLIP AND MATCH WITH MATERIAL
FROM HOLDING AREA
MATERIAL
TITLE II CARD EXPECTED
FROM TP.MAR (MONTHLY)
GN.SYS.OL01
OUT OF DATE STANDING SEARCH REQUESTS LISTING
OPERATION CT.DIS.10.01
PULL MATERIAL FROM HOLDING AREA
PROCEDURE CT.DIS.12
KEY MARC SEARCH
MARC DATA EXPECTED
LISTING FOR MUSIC SENT TO MUSIC CATALOGING
TO MUSIC CATALOGING

SHEET FIVE



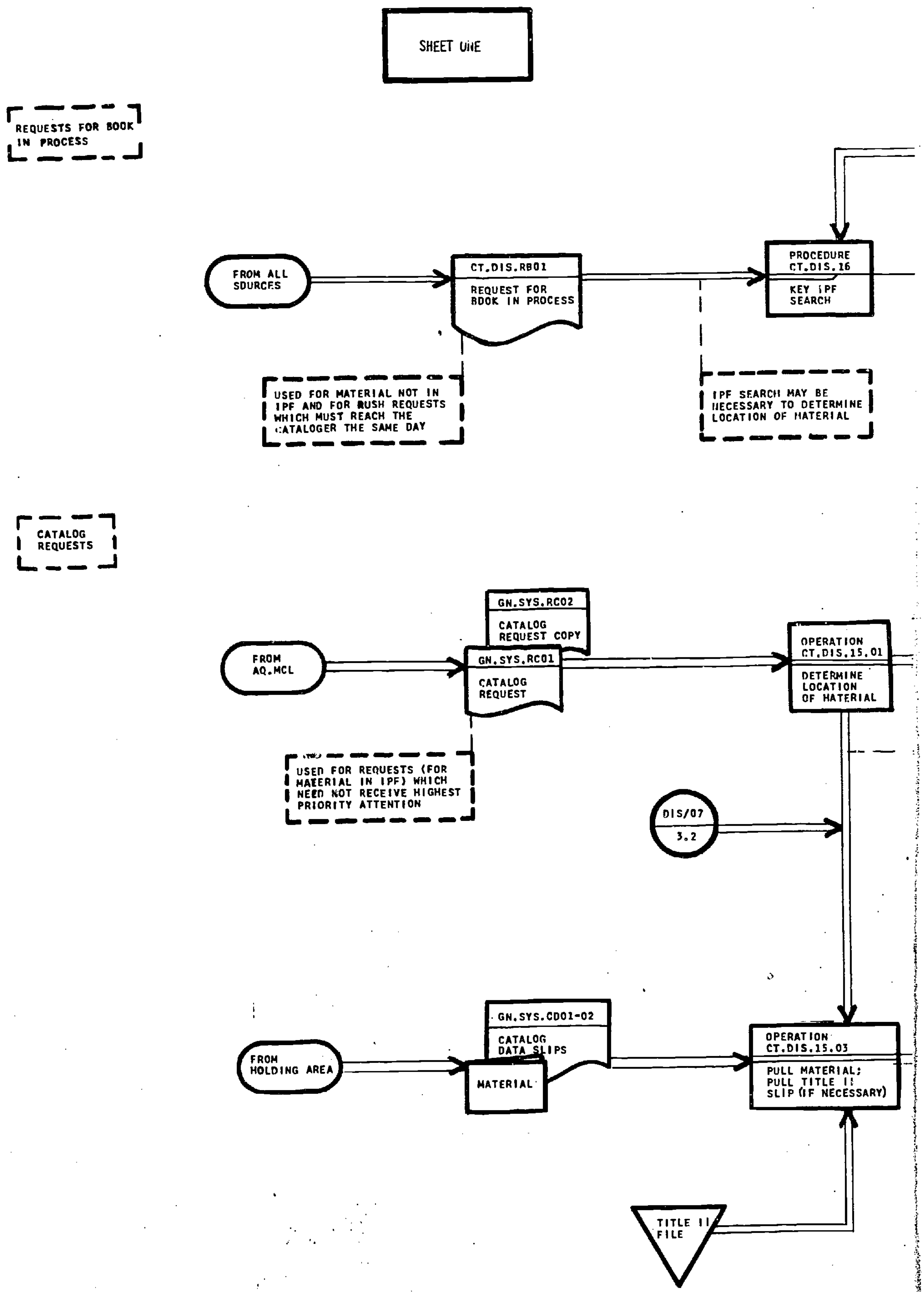
STANFORD UNIVERSITY LIBRARIES

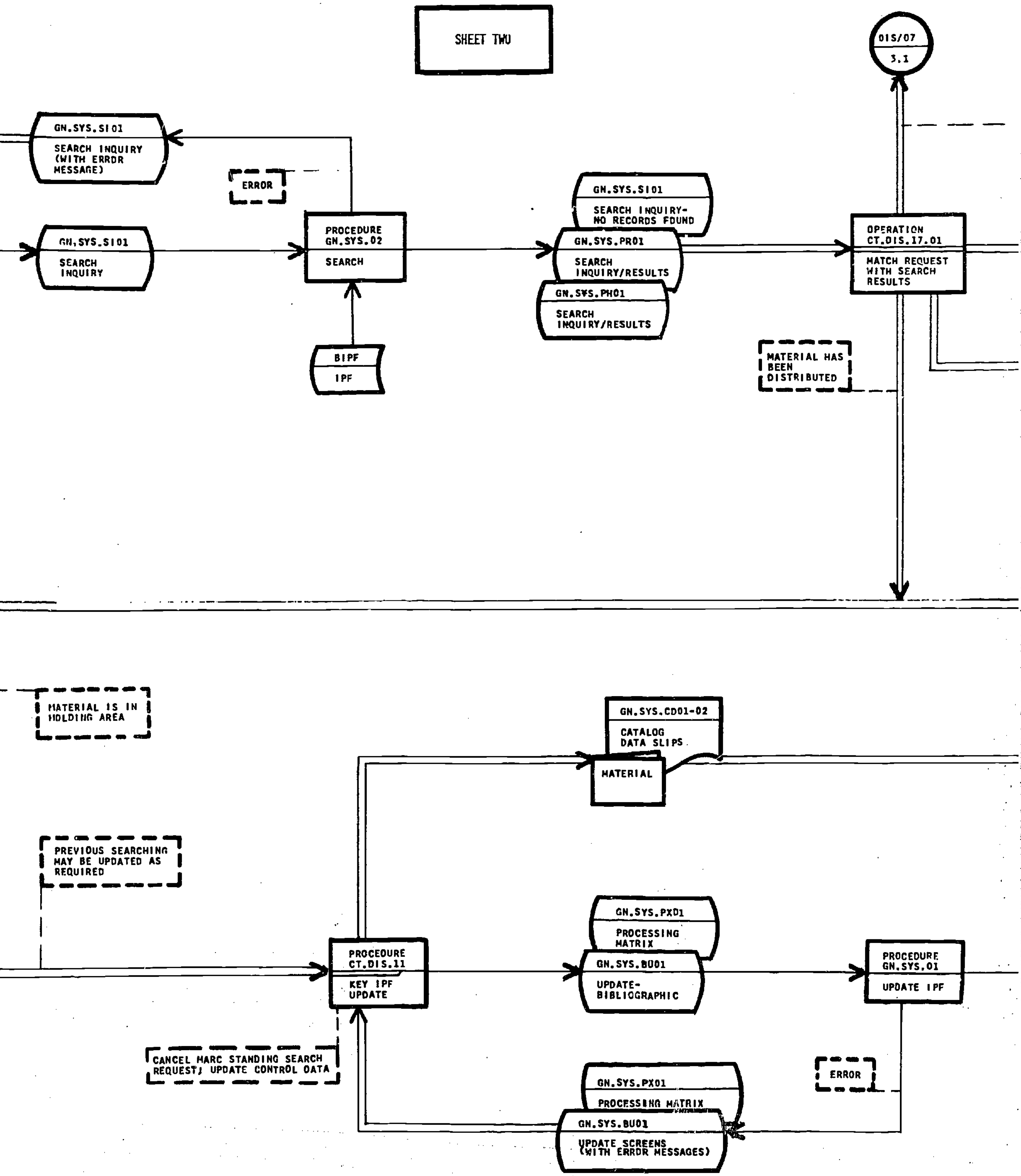
APPROVAL(S) REQUIRED: J. HITCHCOCK

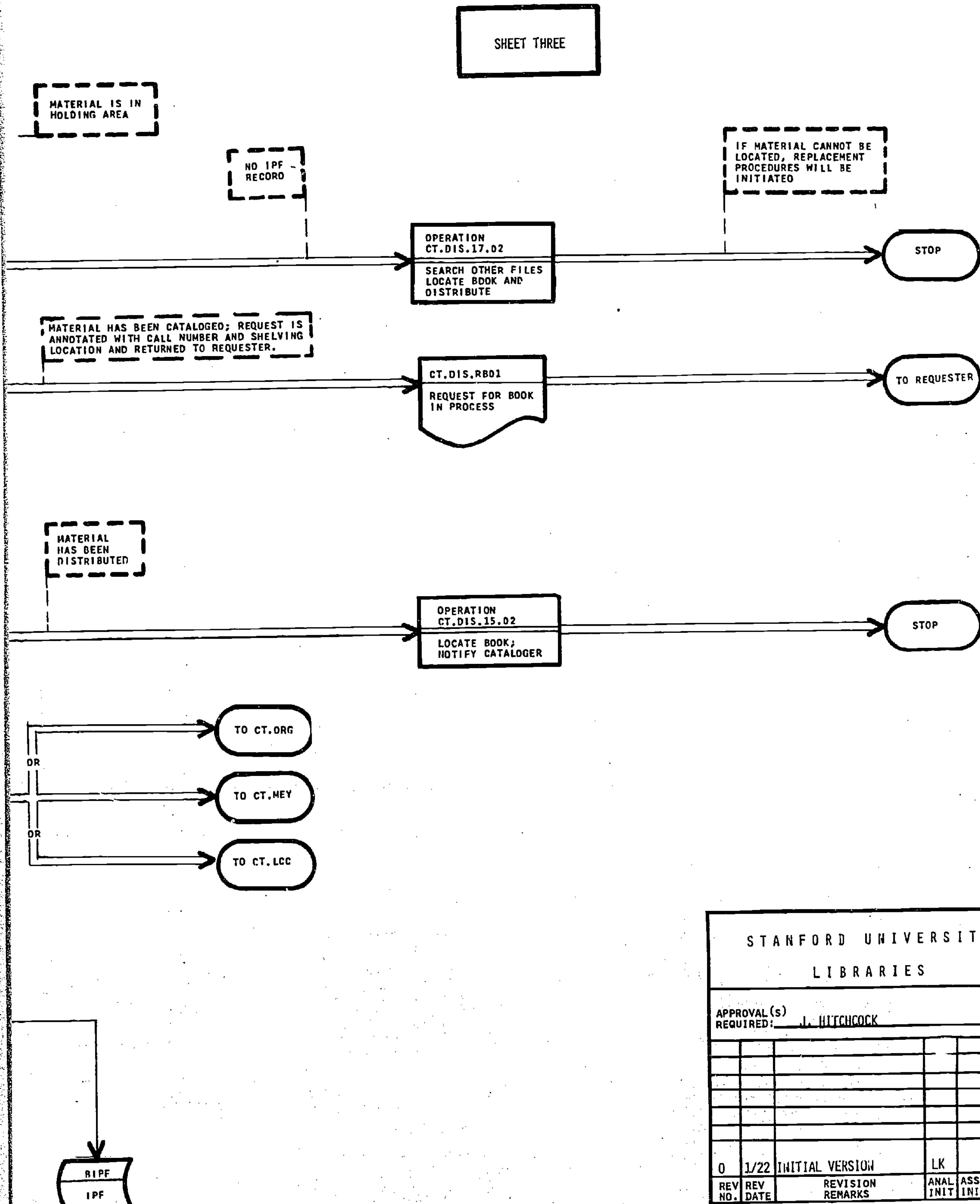
REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEPT INIT
0	1/22	INITIAL VERSION	LK		

TITLE: DISTRIBUTION AND MARC SEARCHING PROCESS

ID: CT.DIS


SHEET UNE
REQUESTS FOR BOOK IN PROCESS
FROM ALL SOURCES
CT.DIS.RB01
REQUEST FOR BOOK IN PROCESS
PROCEDURE CT.DIS.16
KEY IPF SEARCH
USED FOR MATERIAL NOT IN IPF AND FOR RUSH REQUESTS WHICH MUST REACH THE CATALOGER THE SAME DAY
IPF SEARCH MAY BE NECESSARY TO DETERMINE LOCATION OF MATERIAL
CATALOG REQUESTS
GN.SYS.RC02
CATALOG REQUEST COPY
FROM AQ.MCL
GN.SYS.RC01
CATALOG REQUEST
OPERATION CT.DIS.15.01
DETERMINE LOCATION OF MATERIAL
USED FOR REQUESTS (FOR MATERIAL IN IPF) WHICH NEED NOT RECEIVE HIGHEST PRIORITY ATTENTION
DIS/07
3.2
GN.SYS.CD01-02
CATALOG DATA SLIPS
FROM HOLDING AREA
MATERIAL
OPERATION CT.DIS.15.03
PULL MATERIAL; PULL TITLE II SLIP (IF NECESSARY)
TITLE II FILE


SHEET TWO
DIS/07
3.1
GN.SYS.SI01
SEARCH INQUIRY (WITH ERROR MESSAGE)
ERROR
GN.SYS.SI01
SEARCH INQUIRY-NO RECORDS FOUND
GN,SYS.SI01
SEARCH INQUIRY
PROCEDURE GN.SYS.02
SEARCH
GN.SYS.PR01
SEARCH INQUIRY/RESULTS
OPERATION CT.DIS.17.01
MATCH REQUEST WITH SEARCH RESULTS
GN.SYS.PH01
SEARCH INQUIRY/RESULTS
BIPF
IPF
MATERIAL HAS BEEN DISTRIBUTED
MATERIAL IS IN HOLDING AREA
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
PREVIOUS SEARCHING MAY BE UPDATED AS REQUIRED
GN.SYS.PX01
PROCESSING MATRIX
PROCEDURE CT.DIS.11
KEY IPF UPDATE
GN.SYS.BU01
UPDATE-BIBLIOGRAPHIC
PROCEDURE GN.SYS.01
UPDATE IPF
CANCEL MARC STANDING SEARCH REQUEST; UPDATE CONTROL DATA
ERROR
GN.SYS.PX01
PROCESSING MATRIX
GN.SYS.BU01
UPDATE SCREENS (WITH ERROR MESSAGES)



STANFORD UNIVERSITY LIBRARIES

APPROVAL(s) REQUIRED: J. HITCHCOCK

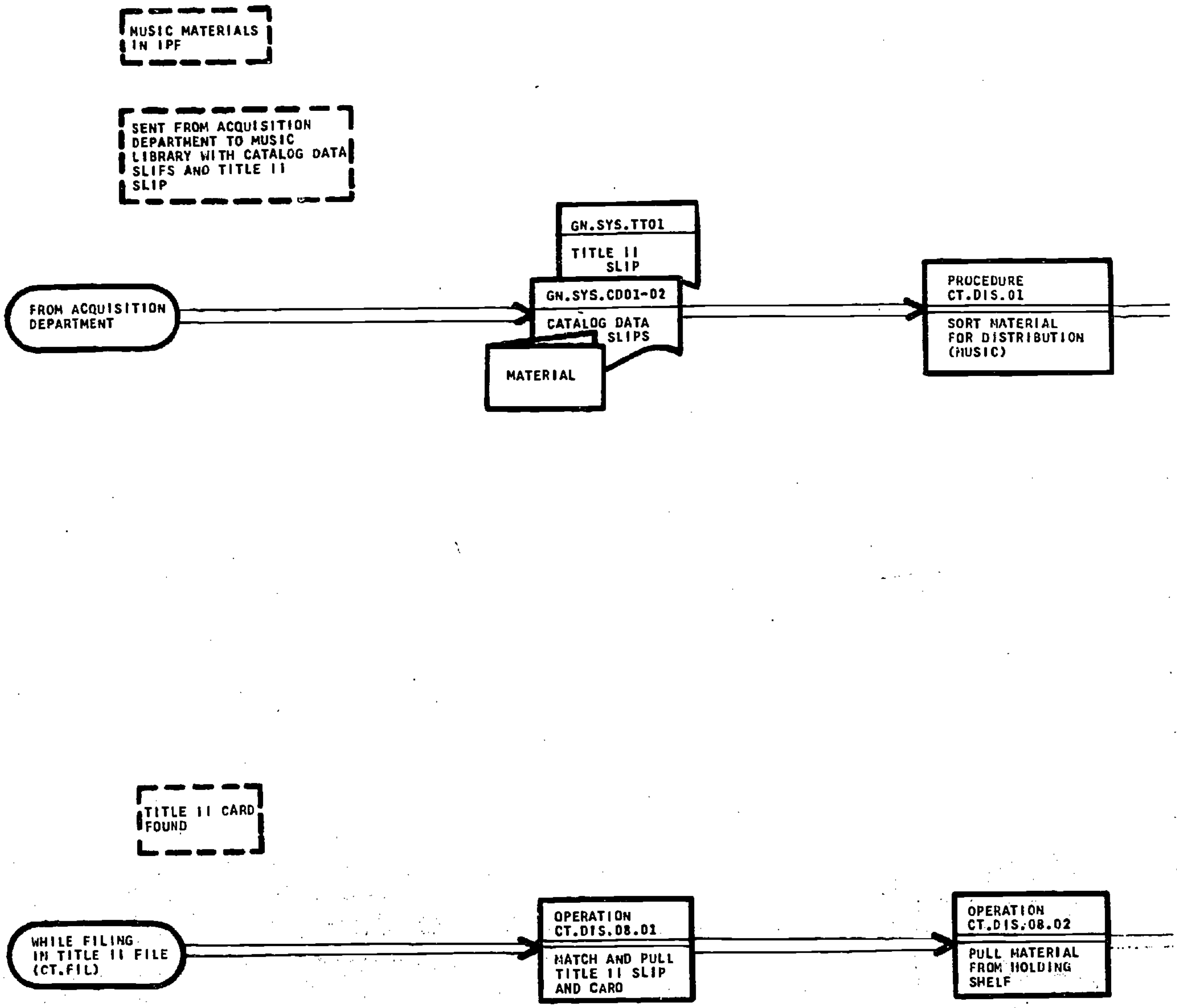
REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEPT INIT
0	1/22	INITIAL VERSION	LK		

TITLE: DISTRIBUTION AND MARC SEARCHING PROCESS

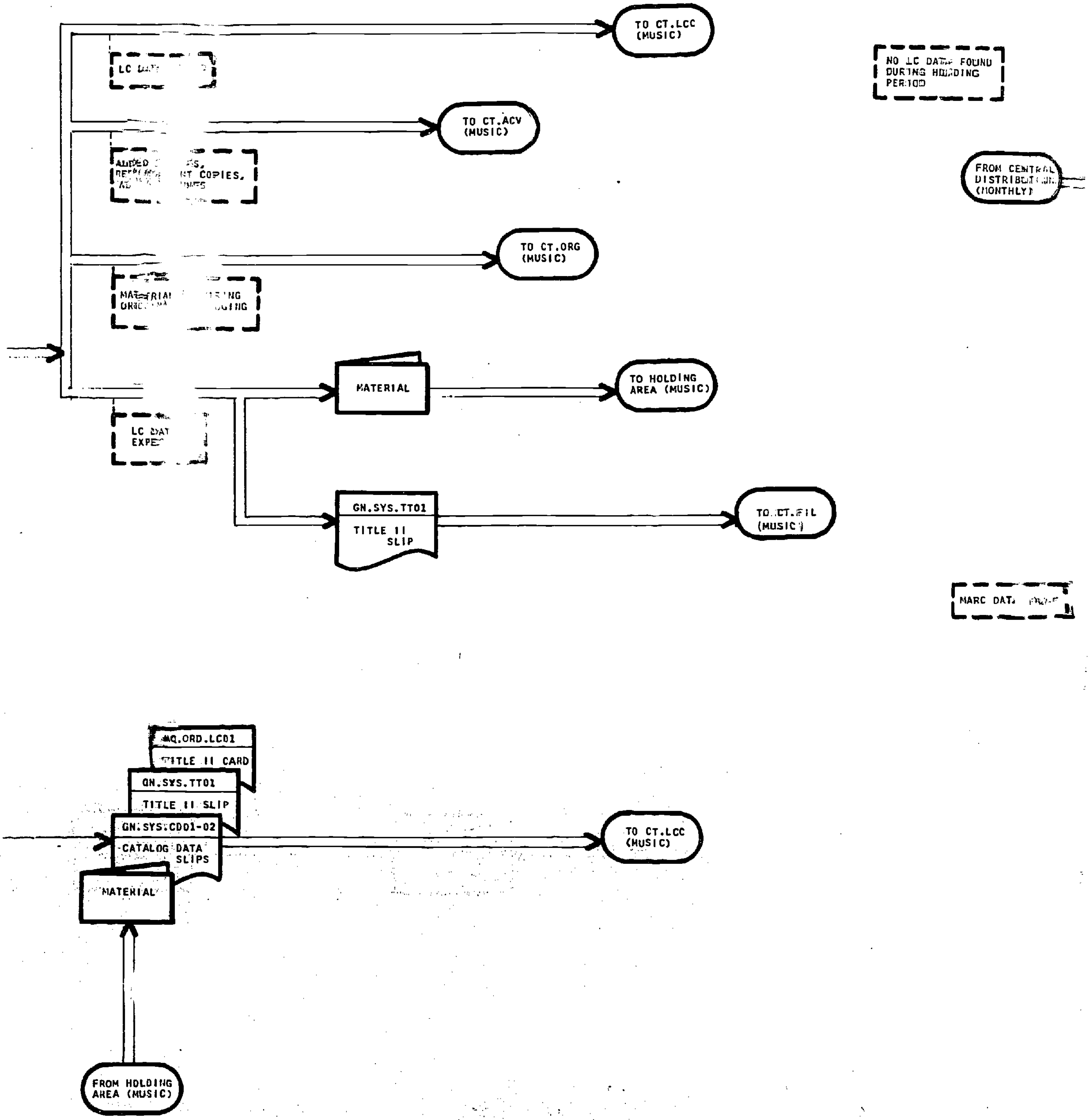
ID: CT.DIS

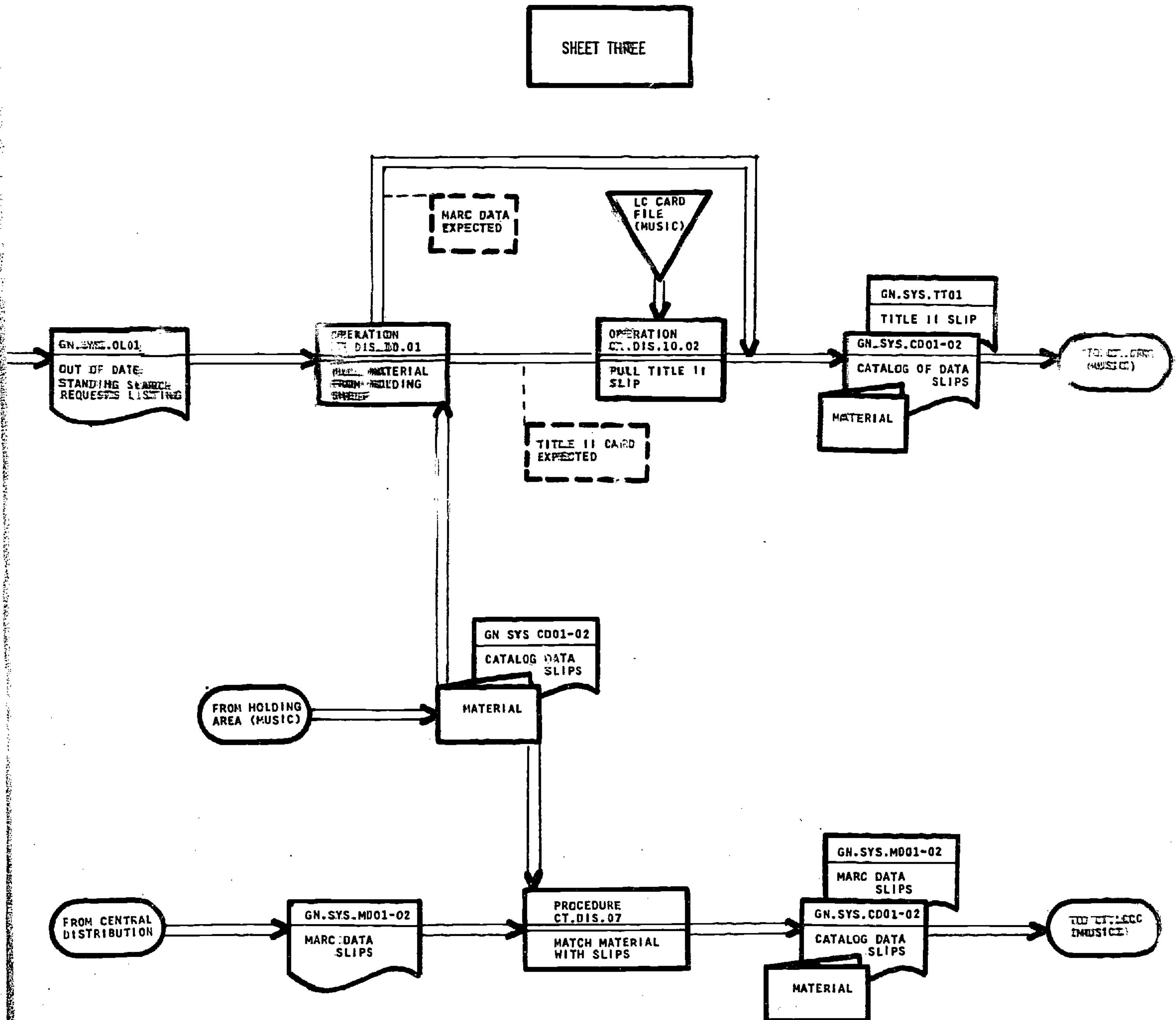
3 SHEETS PAGE 3 OF 5

SHEET ONE


MUSIC MATERIALS IN IPF
SENT FROM ACQUISITION DEPARTMENT TO MUSIC LIBRARY WITH CATALOG DATA SLIFS AND TITLE II SLIP
FROM ACQUISITION DEPARTMENT
GN.SYS.TT01
TITLE II SLIP
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
PROCEDURE CT.DIS.01
SORT MATERIAL FOR DISTRIBUTION (MUSIC)
TITLE II CARD FOUND
WHILE FILING IN TITLE II FILE (CT.FIL)
OPERATION CT.DIS.08.01
MATCH AND PULL TITLE II SLIP AND CARD
OPERATION CT.DIS.08.02
PULL MATERIAL FROM HOLDING SHELF

SHEET TWO


TO CT.LCC (MUSIC)
NO LC DATA FOUND DURING HOLDING PERIOD
TO CT.ACV (MUSIC)
FROM CENTRAL DISTRIBUTION (MONTHLY)
TO CT.ORG (MUSIC)
MATERIAL
TO HOLDING AREA (MUSIC)
GN.SYS.TT01
TITLE II SLIP
TO CT.FIL (MUSIC)
MARC DATA
AQ.ORD.LC01
TITLE II CARD
GN.SYS.TT01
TITLE II SLIP
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
TO CT.LCC (MUSIC)
FROM HOLDING AREA (MUSIC)



STANFORD UNIVERSITY
LIBRARIES

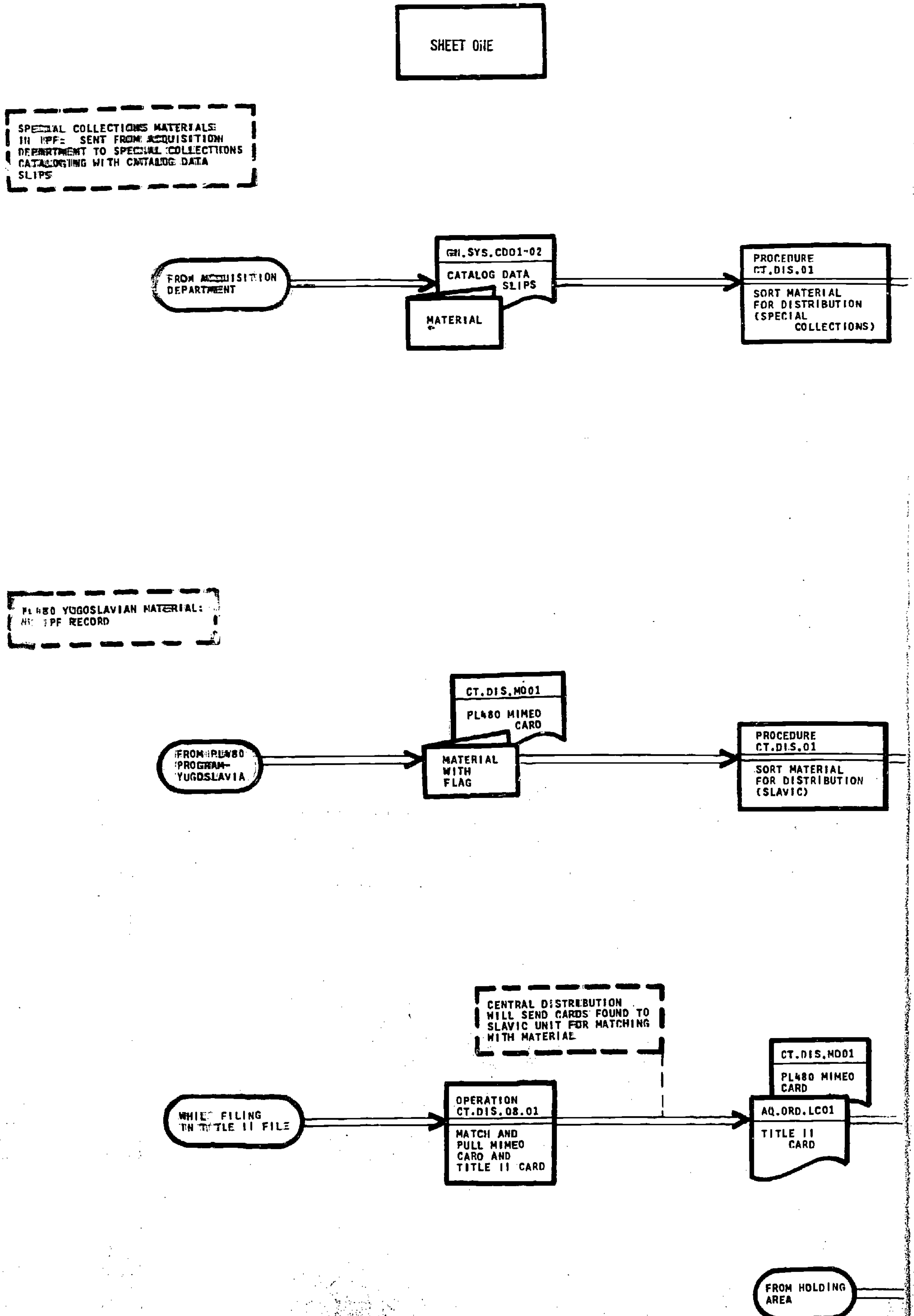
APPROVAL(s) REQUIRED: J. HITCHCOCK

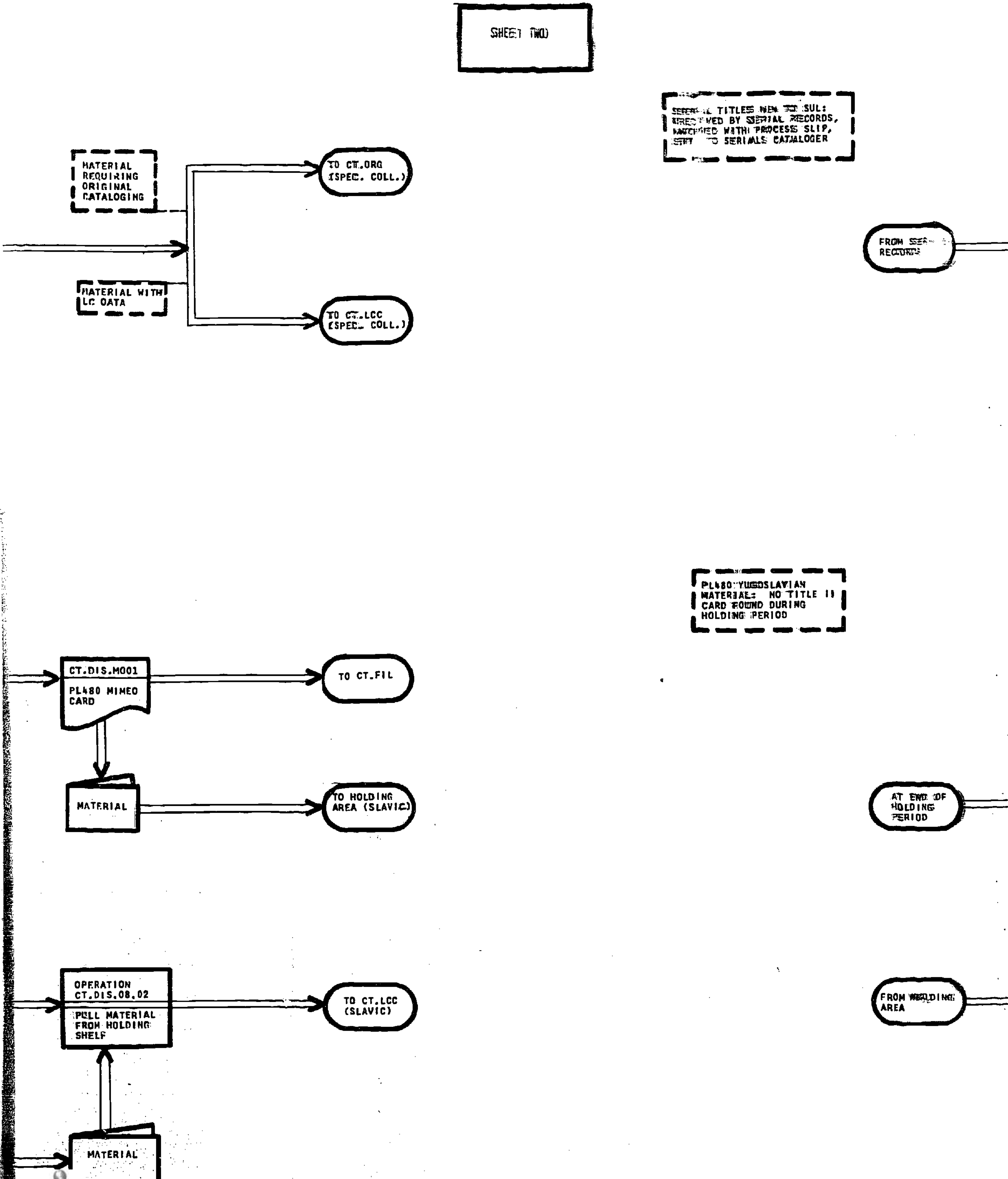
REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEPT INIT
0	1/22	INITIAL VERSION	LK		

TITLE: DISTRIBUTION AND MARC SEARCHING PROCESS

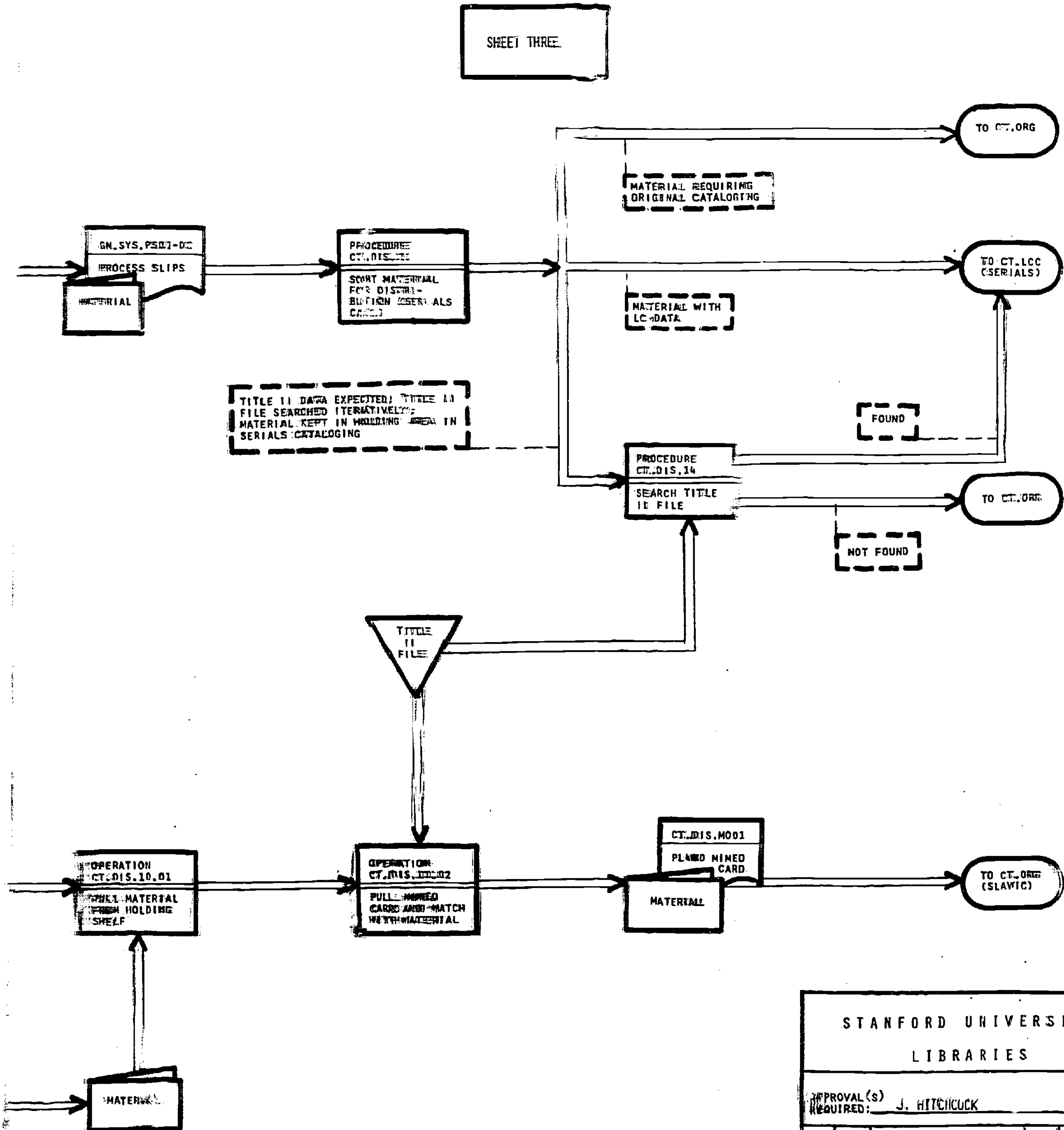
ID: CT.DIS

3 SHEETS PAGE 4 OF 5


SHEET ONE
SPECIAL COLLECTIONS MATERIALS IN IPF: SENT FROM ACQUISITION DEPARTMENT TO SPECIAL COLLECTIONS CATALOGING WITH CATALOG DATA SLIPS
FROM ACQUISITION DEPARTMENT
GN.SYS.CD01-02
CATALOG DATA SLIPS
MATERIAL
PROCEDURE CT.DIS.01
SORT MATERIAL FOR DISTRIBUTION (SPECIAL COLLECTIONS)
PL480 YUGOSLAVIAN MATERIAL: NO IPF RECORD
FROM PL480 PROGRAM-YUGOSLAVIA
CT.DIS.M001
PL480 MIMEO CARD
MATERIAL WITH FLAG
PROCEDURE CT.DIS.01
SORT MATERIAL FOR DISTRIBUTION (SLAVIC)
CENTRAL DISTRIBUTION WILL SEND CARDS FOUND TO SLAVIC UNIT FOR MATCHING WITH MATERIAL
WHILE FILING IN TITLE II FILE
OPERATION CT.DIS.08.01
MATCH AND PULL MIMEO CARD AND TITLE II CARD
CT.DIS.M001
PL480 MIMEO CARD
AQ.ORD.LC01
TITLE II CARD
FROM HOLDING AREA



SHEET TWO
TITLES NEW TO SUL:
BY SERIAL RECORDS,
PROCESS SLIP,
TO SERIALS CATALOGER
MATERIAL REQUIRING ORIGINAL CATALOGING
TO CT.ORG (SPEC. COLL.)
FROM
MATERIAL WITH LC DATA
TO CT.LCC
PL480 YUGOSLAVIAN MATERIAL: NO TITLE II CARD FOUND DURING HOLDING PERIOD
CT.DIS.M001
PL480 MIMEO CARD
TO CT.FIL
MATERIAL
TO HOLDING AREA (SLAVIC)
AT END OF HOLDING PERIOD
OPERATION CT.DIS.08.02
PULL MATERIAL FROM HOLDING SHELF
TO CT.LCC (SLAVIC)
FROM
AREA
MATERIAL



STANFORD UNIVERSITY LIBRARIES

APPROVAL(S) REQUIRED: J. HITCHCOCK

REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT
0	1/22	INITIAL VERSION	LK	

TITLE: DISTRIBUTION AND MARC SEARCHING PROCESS

ID: CT.DIS

3 SHEETS PAGE 5 OF 5

THE MEYER CATALOGING PROCESS (CT.MEY)

The purpose of the Meyer Cataloging process (CT.MEY) is to prepare and maintain catalog records for the material cataloged for the J. Henry Meyer Memorial Library. Except in the distribution of material, this process is separate from the other Catalog Department processes. The separation has proved to be an efficient allocation of personnel since Meyer uses different cataloging rules and formats, and since maintaining a catalog in book form necessitates different approaches and techniques.

Most Meyer material will be received in the Catalog Department through the Distribution Process (CT.DIS). Material for which an In Process File (IPF) record has been created already and for which MARC data is expected will be placed in the Catalog Department holding area, to be held no longer than two months. During the holding period the titles will be searched automatically against each weekly MARC tape; when matches are discovered, CT.DIS will forward the material to CT.MEY. The MARC records that are transferred to the IPF as a result of this matching are also printed out on two-part MARC Data Slips (GN.SYS.MD01-02), and inserted in the material before it is forwarded.

Material for which no MARC data is expected, or with which no MARC data has been matched after two months, will be sent to CT.MEY with a printout of the IPF record on a two-part Catalog Data Slip (GN.SYS.CD01-02).

Material ordered before the implementation of the Acquisition Subsystem and received after the implementation of CT.MEY will not have IPF records. Any such item will be sent to CT.MEY accompanied by an SUL-25 form (AQ.ORD.BR01) and/or the green copy from the SUL-7 form (CT.DIS.SU01). In time this category of material will diminish and eventually disappear.

Gifts selected by the Meyer reference staff for addition to the undergraduate library will be received in CT.MEY with two-part Catalog Data Slips, since the Gift Division will have created the requisite IPF records.

For most books intended for Meyer Reserves, an IPF record will already have been created by the Order Division before the books are received in CT.MEY. A few books--which are needed on such short notice that there is not time for Order Division processing--will be received in CT.MEY without IPF records.

Meyer phonodiscs and tapes will continue to go to the Audio Library. After a sequential number is assigned there, the material will be transferred to CT.MEY for cataloging. The Order Division will not enter phonodiscs or tapes into the BIPF.

After material has been received in CT.MEY, it will be sorted (CT.MEY.01) according to cataloging priorities and source of bibliographic data. After sorting and prior to cataloging, all material will be searched in the Meyer Inventory File (INV); and depending on the source of the bibliographic data on the accompanying process slips, different data elements will be checked.

If the source of the bibliographic data for new material being searched is the Library of Congress (either captured from MARC or keyed in by the Order Division from a Title II LC depository card or an LC printed catalog entry), then the entry, call number, subject and other added entries can be checked against the INV for compatibility. If the source of the bibliographic data is not LC, then only the designated entry or alternative entries can be checked for compatibility with the INV.

Items marked as added volumes or copies on their process slips will be checked against the corresponding INV record. If this search shows that the item is an added volume or copy, it will be sent to the added copy/volume procedure. If searching shows that an item is not an added copy or volume as indicated on its slips, it will be treated as new material.

If the search shows that a supposedly new item is actually an added volume, it will be sent to the added copy/volume procedure. If the search shows that a new item is an unexpected duplicate, the Meyer reference staff will be asked it they want it added to the collection. If they do, it will be sent for added copy/volume processing. If the item is not wanted, it will be returned to the Order Division.

Following the intial search in the INV, material will be sent to one of four cataloging procedures:

1. Added Copy/Volume processing (CT.MEY.07). Material to be added to existing holdings, excluding unwanted, unexpected duplicates.

2. LC Cataloging (CT.MEY.04). Material with IPF records created from LC information.

3. Original Cataloging (CT.MEY.05). Material with IPF records created from other than LC information.

4. Cataloging without INV or IPF records (CT.MEY.12). Material ordered under the manual system and therefore without IPF records. Also will cover rush Meyer reserve material not processed by the Order Division.

In added copy/volume processing (CT.MEY.07), a copy (This can be done immediately after the initial search by keying a transfer command on the search results screen.) The update is then keyed into the IPF, together with a command to copy the updated record back into the INV. As a result the INV will be updated. Spine labels, a book card, and a Meyer short card also will be generated.

During LC Cataloging, Original Cataloging, or cataloging without INV or IPF records, any of the system files may be searched for information. The IPF can be queried for information about the status of related items. The INV file can be checked to see how personal, corporate, or conference names and subject headings have been established and if an assigned call number is unique to the INV. (This step will replace the searching currently done in the Meyer manual authority files and the shelf list.) The Catalog Data File (CDF) can be searched to see if a title has already been cataloged for the main library. The MARC (MRC) file can be searched to see if a matching MARC record is available.

If a matching record is found, it can be copied to the IPF for updating. If the source of the bibliographic data in this record is LC, the material is routed to the LC Cataloging procedure (CT.MEY.04). If the source of the bibliographic data is other than LC, the material is routed to the Original Cataloging procedure (CT.MEY.05).

If an IPF record exists, it will be updated in cataloging procedures CT.MEY.04 and CT.MEY.05. If an IPF record does not exist, then the entire bibliographic record must be keyed into the IPF in cataloging procedure CT.MEY.12.

As the result of an update or input and a copy command, the INV will be updated and spine labels, book cards, and Meyer short cards will be generated. The spine labels and book card will be sent to End Processing for matching with the material. The short card will be sent directly to the Meyer Library.

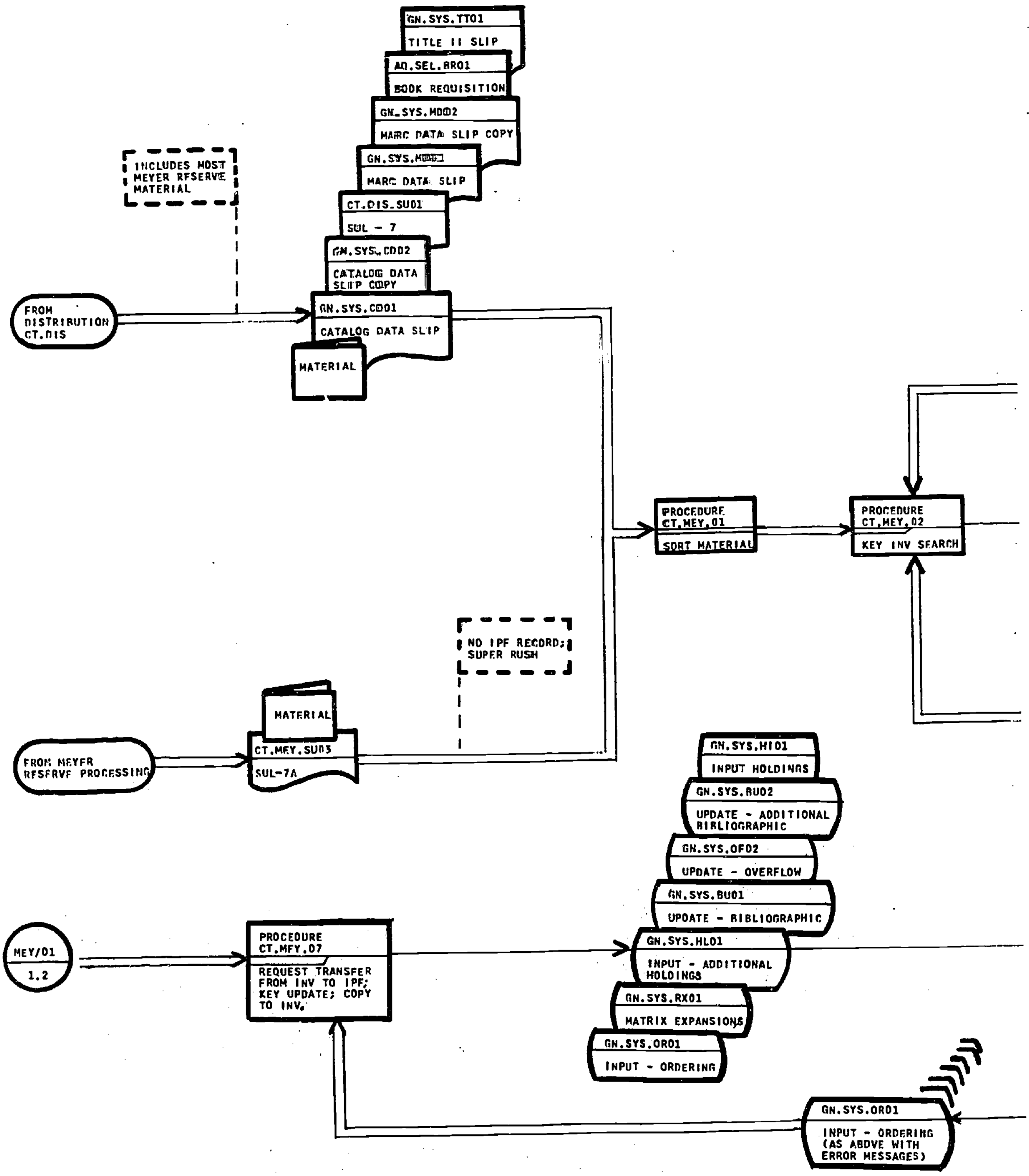
As the final action with the material in all of the cataloging procedures, the call number will be written in the book. Then, depending on whether or not a binding decision is needed, the material is sent either to the Meyer Library or to Binding and Finishing. Neither book cards nor spine labels will be generated for items sent tothe Meyer Library; instead, the IPF will be updated to show that the item has been sent for a binding decision. If the material is returned to the Meyer Library for binding in End Processing, then spine labels and book cards will be generated.

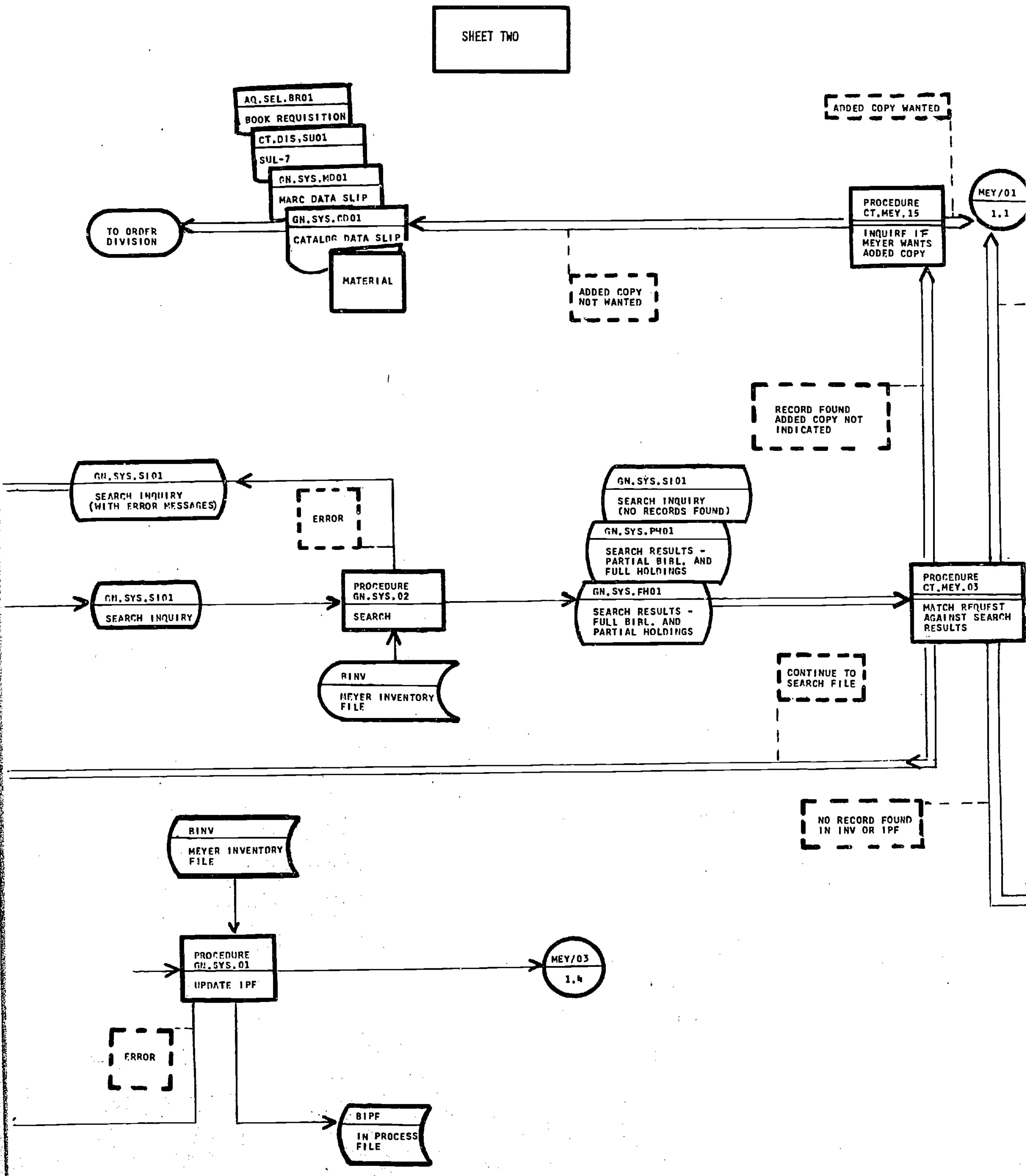
Audio material will be transferred from the Audio Library to CT.MEY without previously created IPF records. This INV will be searched first to see if there are any added copies. If so, they will be returned to th Audio Library. During the Audio Cataloging procedure (CT.MEY.14), any of the system files can be queried. (Until MARC expands to include phonodiscs and tapes, this search capability will be limited mainly to checking the INV as an authority for audio cataloging.) The bibliographic data then will be keyed into the IPF, resulting in an updated INV and an Audio Short Card that is sent directly to the Audio Library.

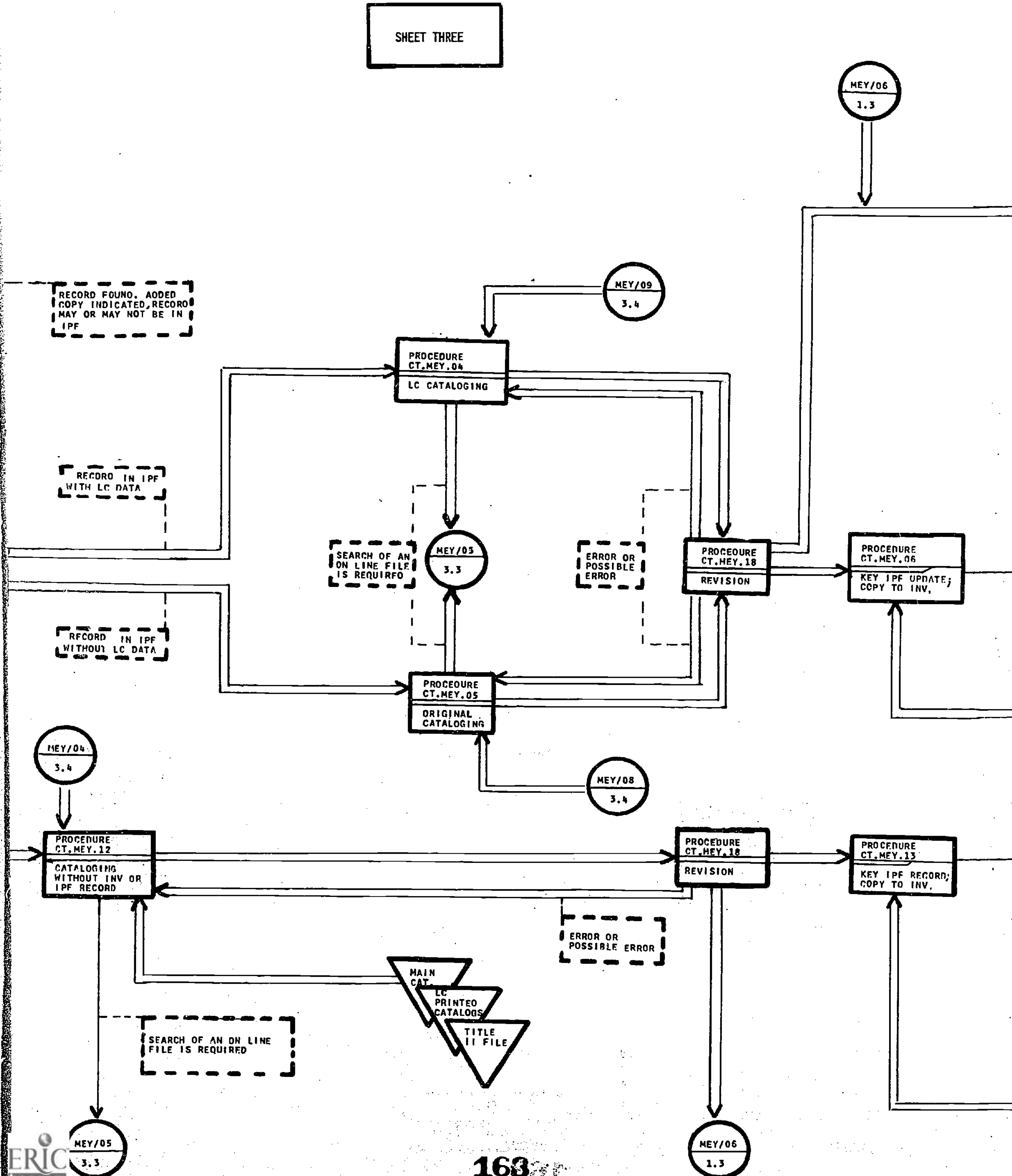
Any maintenance of an existing INV record will be accomplished by copying the record into the IPF, keying the update, and moving the updated record back to the INV. Replacement spine labels, book cards, or short cards can be generated as needed. Records can also be deleted from the INV in the maintenance function (CT.MEY.19).

Material that Meyer Library has sent for commercial binding will be returned to CT.MEY. For a monograph, spine labels and a book card will be generated as needed, and the IPF record will be updated to show that the item has gone to Binding and Finishing. For a bound serial volume, the INV record will be transferred to the IPF so that the holdings can be updated. Any necessary spine labels or book card can also be generated.

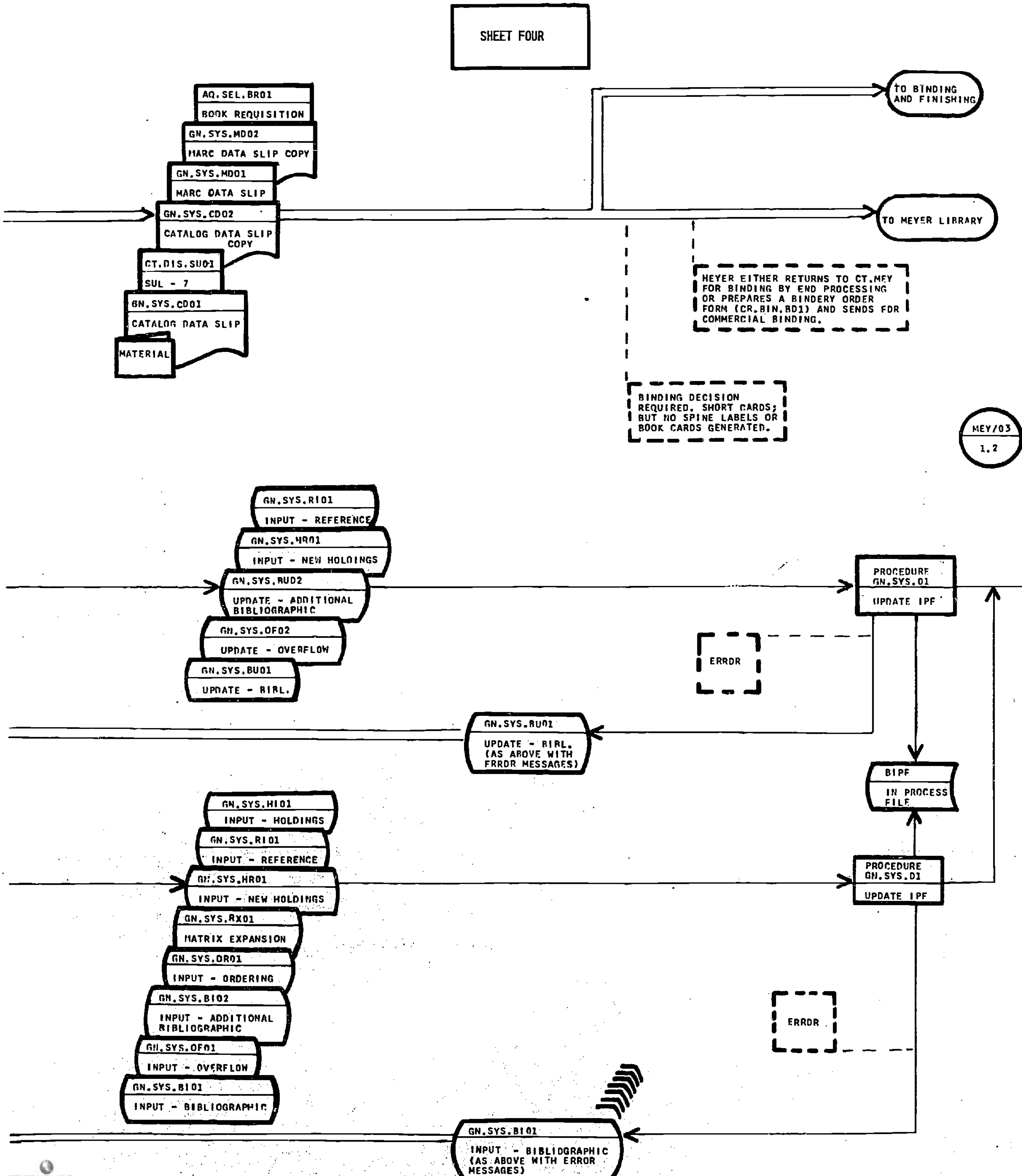
SHEET ONE


GN.SYS.TT01
TITLE II SLIP
AQ.SEL.BR01
BOOK REQUISITION
GN.SYS.MD02
MARC DATA SLIP COPY
GN.SYS.MD01
MARC DATA SLIP
CT.DIS.SU01
SUL - 7
GN.SYS.CD02
CATALOG DATA SLIP COPY
GN.SYS.CD01
CATALOG DATA SLIP
MATERIAL
INCLUDES MOST MEYER RESERVE MATERIAL
FROM DISTRIBUTION CT.DIS
PROCEDURE CT.MEY.01
SORT MATERIAL
PROCEDURE CT.MEY.02
KEY INV SEARCH
NO IPF RECORD; SUPER RUSH
MATERIAL
CT.MEY.SU03
SUL-7A
FROM MEYER RESERVE PROCESSING
GN.SYS.HI01
INPUT HOLDINGS
GN.SYS.BU02
UPDATE - ADDITIONAL BIBLIOGRAPHIC
GN.SYS.OF02
UPDATE - OVERFLOW
GN.SYS.BU01
UPDATE - BIBLIOGRAPHIC
GN.SYS.HL01
INPUT - ADDITIONAL HOLDINGS
GN.SYS.RX01
MATRIX EXPANSIONS
GN.SYS.OR01
INPUT - ORDERING
MEY/01
1.2
PROCEDURE CT.MEY.07
REQUEST TRANSFER FROM INV TO IPF; KEY UPDATE; COPY TO INV.
GN.SYS.OR01
INPUT - ORDERING (AS ABOVE WITH ERROR MESSAGES)

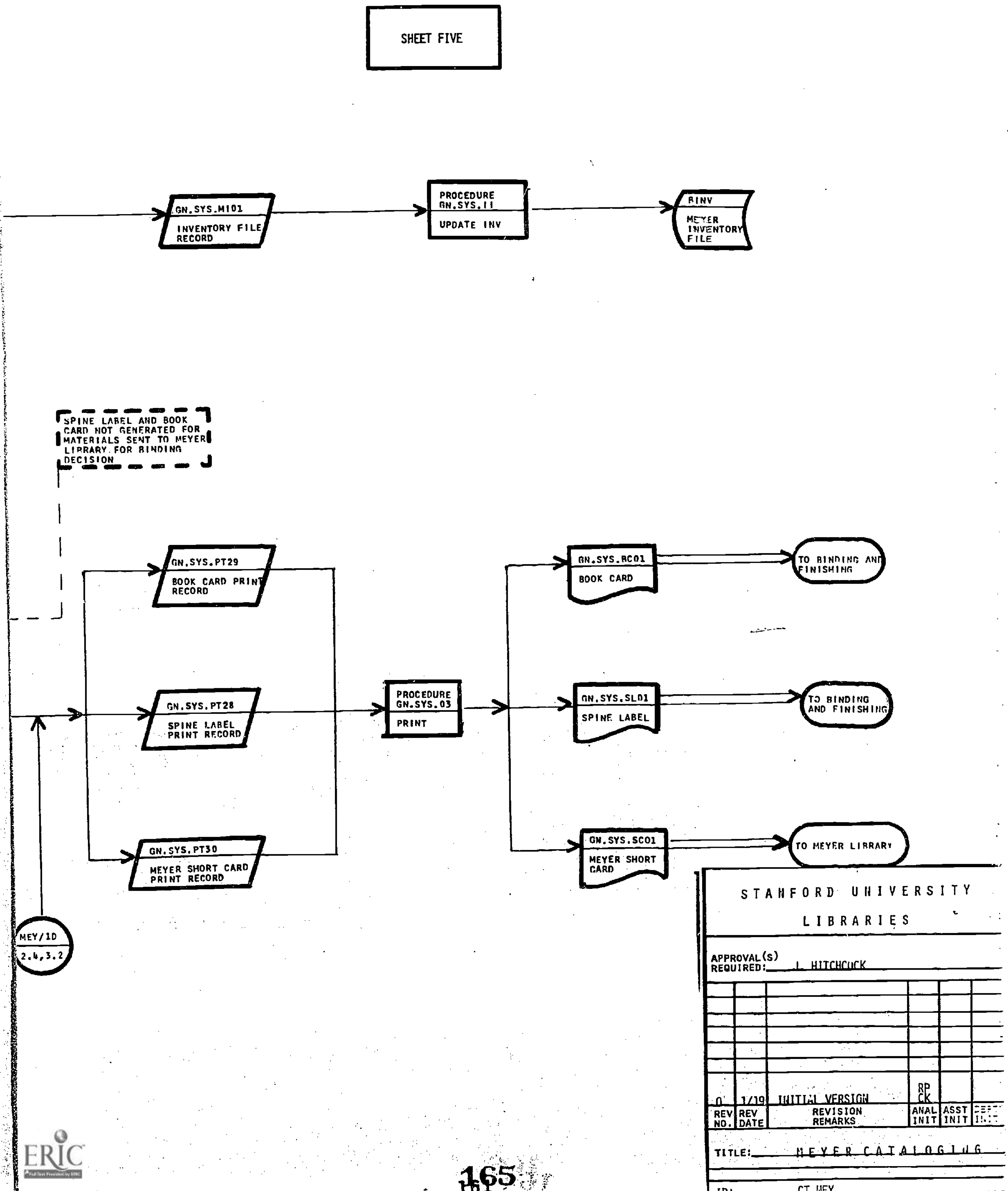

SHEET TWO
AQ.SEL.BR01
BOOK REQUISITION
CT.DIS.SU01
SUL-7
GN.SYS.MD01
MARC DATA SLIP
GN.SYS.CD01
CATALOG DATA SLIP
MATERIAL
TO ORDER DIVISION
ADDED COPY WANTED
ADDED COPY NOT WANTED
PROCEDURE CT.MEY.15
INQUIRE IF MEYER WANTS ADDED COPY
MEY/01
1.1
RECORD FOUND ADDED COPY NOT INDICATED
GN.SYS.SI01
SEARCH INQUIRY (WITH ERROR MESSAGES)
ERROR
GN.SYS.SI01
SEARCH INQUIRY (NO RECORDS FOUND)
GN.SYS.PH01
SEARCH RESULTS - PARTIAL BIBL. AND FULL HOLDINGS
GN.SYS.FH01
SEARCH RESULTS - FULL BIBL. AND PARTIAL HOLDINGS
GN.SYS.SI01
SEARCH INQUIRY
PROCEDURE GN.SYS.02
SEARCH
PROCEDURE CT.MEY.03
MATCH REQUEST AGAINST SEARCH RESULTS
BINV
MEYER INVENTORY FILE
CONTINUE TO SEARCH FILE
NO RECORD FOUND IN INV OR IPF
BINV
MEYER INVENTORY FILE
PROCEDURE GN.SYS.01
UPDATE IPF
MEY/03
1.4
ERROR
BIPF
IN PROCESS FILE



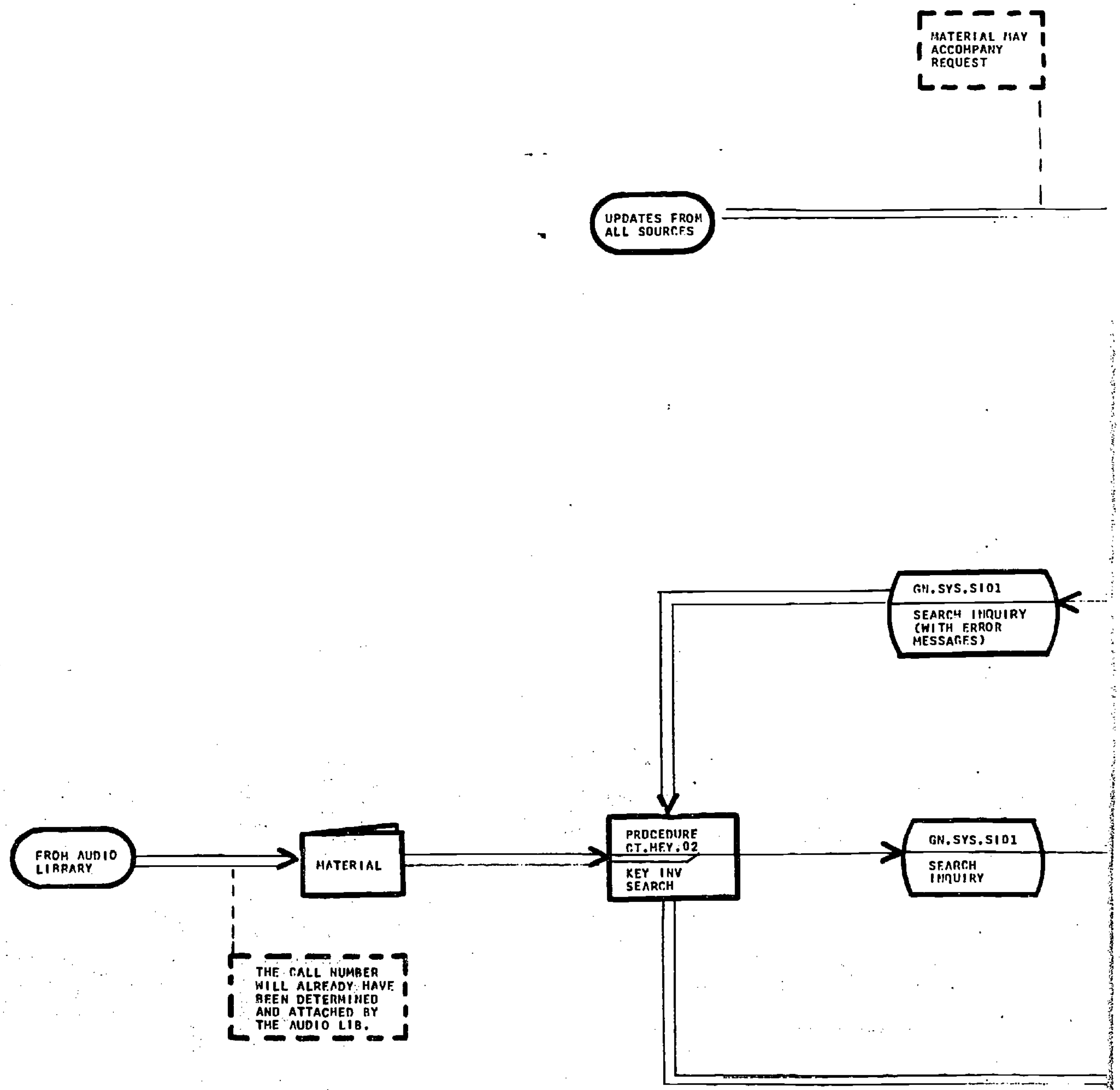
SHEET THREE
MEY/06
1.3
RECORD FOUND, ADDED COPY INDICATED, RECORD MAY OR MAY NOT BE IN IPF
MEY/09
3.4
PROCEDURE CT.MEY.04
LC CATALOGING
RECORD IN IPF WITH LC DATA
SEARCH OF AN ON LINE FILE IS REQUIRED
MEY/05
3.3
ERROR OR POSSIBLE ERROR
PROCEDURE CT.MEY.18
REVISION
PROCEDURE CT.MEY.06
KEY IPF UPDATE; COPY TO INV.
RECORD IN IPF WITHOUT LC DATA
PROCEDURE CT.MEY.05
ORIGINAL CATALOGING
MEY/04
3.4
MEY/08
3.4
PROCEDURE CT.MEY.12
CATALOGING WITHOUT INV OR IPF RECORD
PROCEDURE CT.MEY.18
REVISION
PROCEDURE CT.MEY.13
KEY IPF RECORD; COPY TO INV.
ERROR OR POSSIBLE ERROR
MAIN CAT.
LC PRINTED CATALOGS
TITLE II FILE
SEARCH OF AN ON LINE FILE IS REQUIRED
MEY/05
3.3
MEY/06
1.3



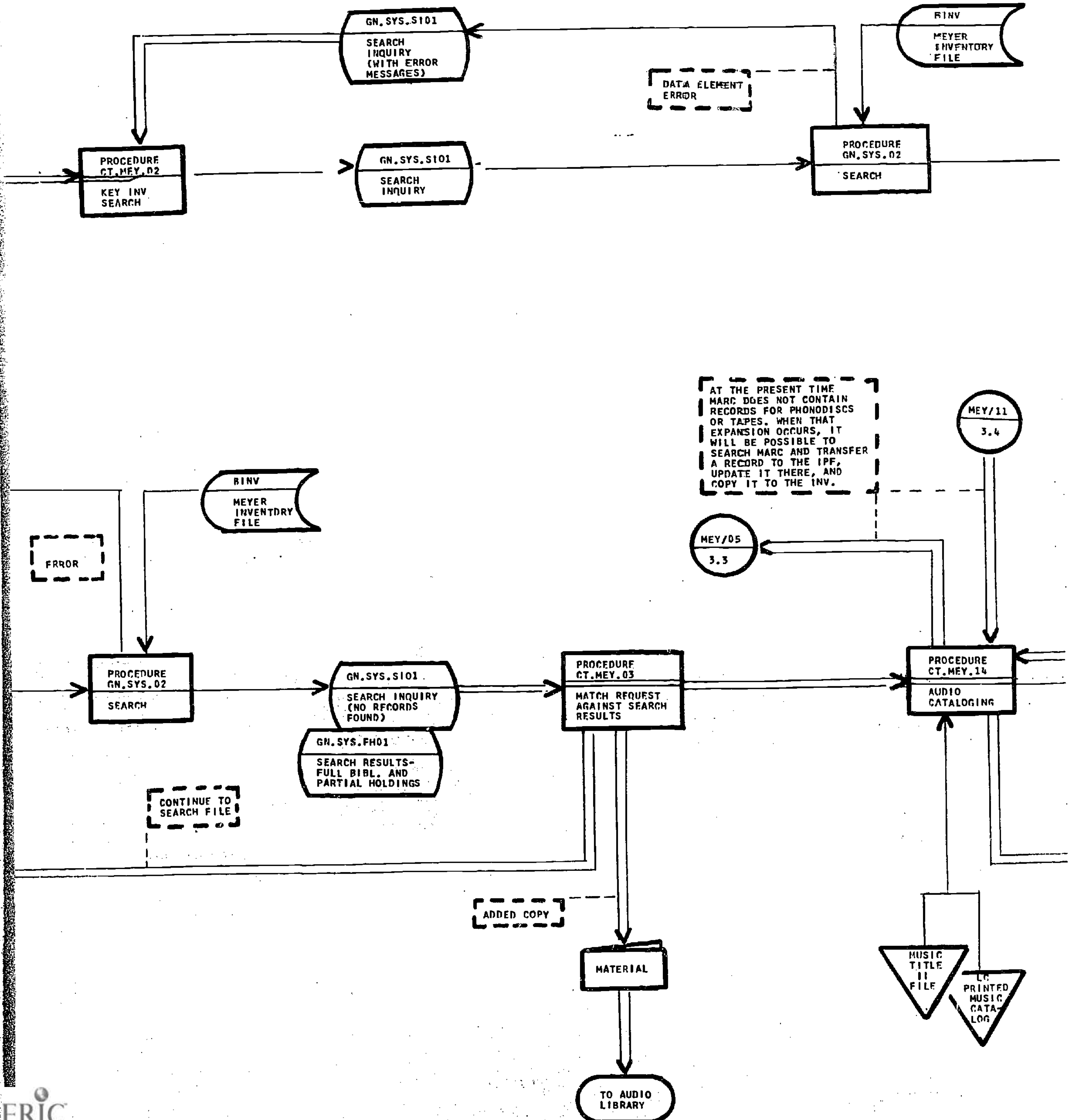
SHEET FOUR
AQ.SEL.BR01
BOOK REQUISITION
GN.SYS.MD02
MARC DATA SLIP COPY
GN.SYS.MD01
MARC DATA SLIP
GN.SYS.CD02
CATALOG DATA SLIP COPY
CT.DIS.SU01
SUL - 7
GN.SYS.CD01
CATALOG DATA SLIP
MATERIAL
TO BINDING AND FINISHING
TO MEYER LIBRARY
MEYER EITHER RETURNS TO CT.MEY FOR BINDING BY END PROCESSING OR PREPARES A BINDERY ORDER FORM (CR.BIN.BD1) AND SENDS FOR COMMERCIAL BINDING.
BINDING DECISION REQUIRED. SHORT CARDS; BUT NO SPINE LABELS OR BOOK CARDS GENERATED.
MEY/03
1.2
GN.SYS.RI01
INPUT - REFERENCE
GN.SYS.HR01
INPUT - NEW HOLDINGS
GN.SYS.BU02
UPDATE - ADDITIONAL BIBLIOGRAPHIC
GN.SYS.OF02
UPDATE - OVERFLOW
GN.SYS.BU01
UPDATE - BIBL.
PROCEDURE GN.SYS.01
UPDATE IPF
ERROR
GN.SYS.BU01
UPDATE - BIBL. (AS ABOVE WITH ERROR MESSAGES)
BIPF
IN PROCESS FILE
GN.SYS.HI01
INPUT - HOLDINGS
GN.SYS.RI01
INPUT - REFERENCE
GN.SYS.HR01
INPUT - NEW HOLDINGS
GN.SYS.RX01
MATRIX EXPANSION
GN.SYS.OR01
INPUT - ORDERING
GN.SYS.BI02
INPUT - ADDITIONAL BIBLIOGRAPHIC
GN.SYS.OF01
INPUT - OVERFLOW
GN.SYS.BI01
INPUT - BIBLIOGRAPHIC
PROCEDURE GN.SYS.01
UPDATE IPF
ERROR
GN.SYS.BI01
INPUT - BIBLIOGRAPHIC (AS ABOVE WITH ERROR MESSAGES)

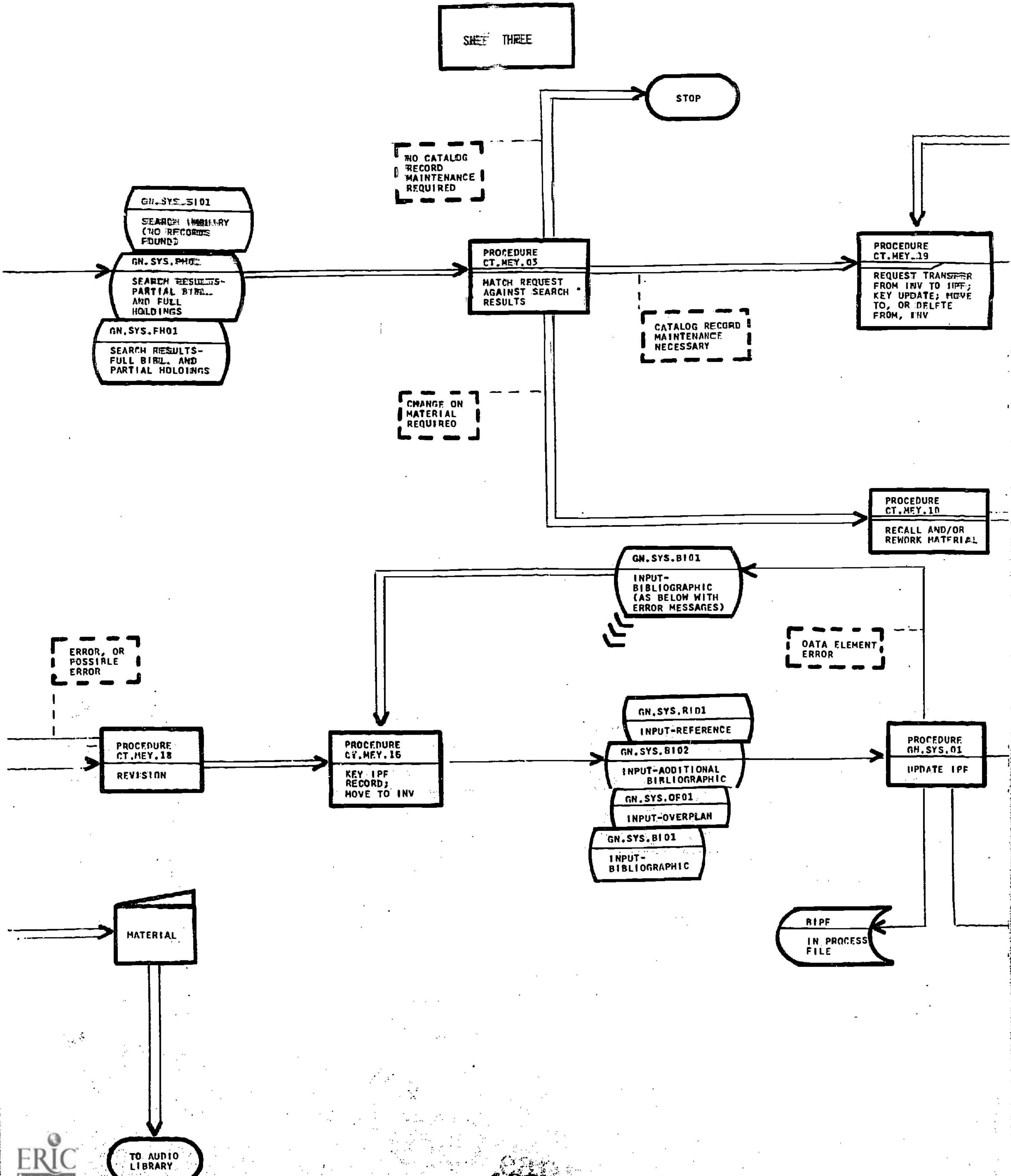

SHEET FIVE
GN.SYS.MI01
INVENTORY FILE RECORD
PROCEDURE GN.SYS.11
UPDATE INV
BINV
MEYER INVENTORY FILE
SPINE LABEL AND BOOK CARD NOT GENERATED FOR MATERIALS SENT TO MEYER LIBRARY FOR BINDING DECISION
GN.SYS.PT29
BOOK CARD PRINT RECORD
GN.SYS.PT28
SPINE LABEL PRINT RECORD
GN.SYS.PT30
MEYER SHORT CARD PRINT RECORD
PROCEDURE GN.SYS.03
PRINT
GN.SYS.BC01
BOOK CARD
GN.SYS.SL01
SPINE LABEL
GN.SYS.SC01
MEYER SHORT CARD
TO BINDING AND FINISHING
TO BINDING AND FINISHING
TO MEYER LIBRARY
MEY/1D
2.4,3.2
STANFORD UNIVERSITY
LIBRARIES
APPROVAL(S) REQUIRED: L. HITCHCOCK
0 1/19 INITIAL VERSION RP CK
REV NO. REV DATE REVISION REMARKS ANAL INIT ASST INIT DEPT INIT
TITLE: MEYER CATALOGING
ID: CT.MEY

SHEET ONE

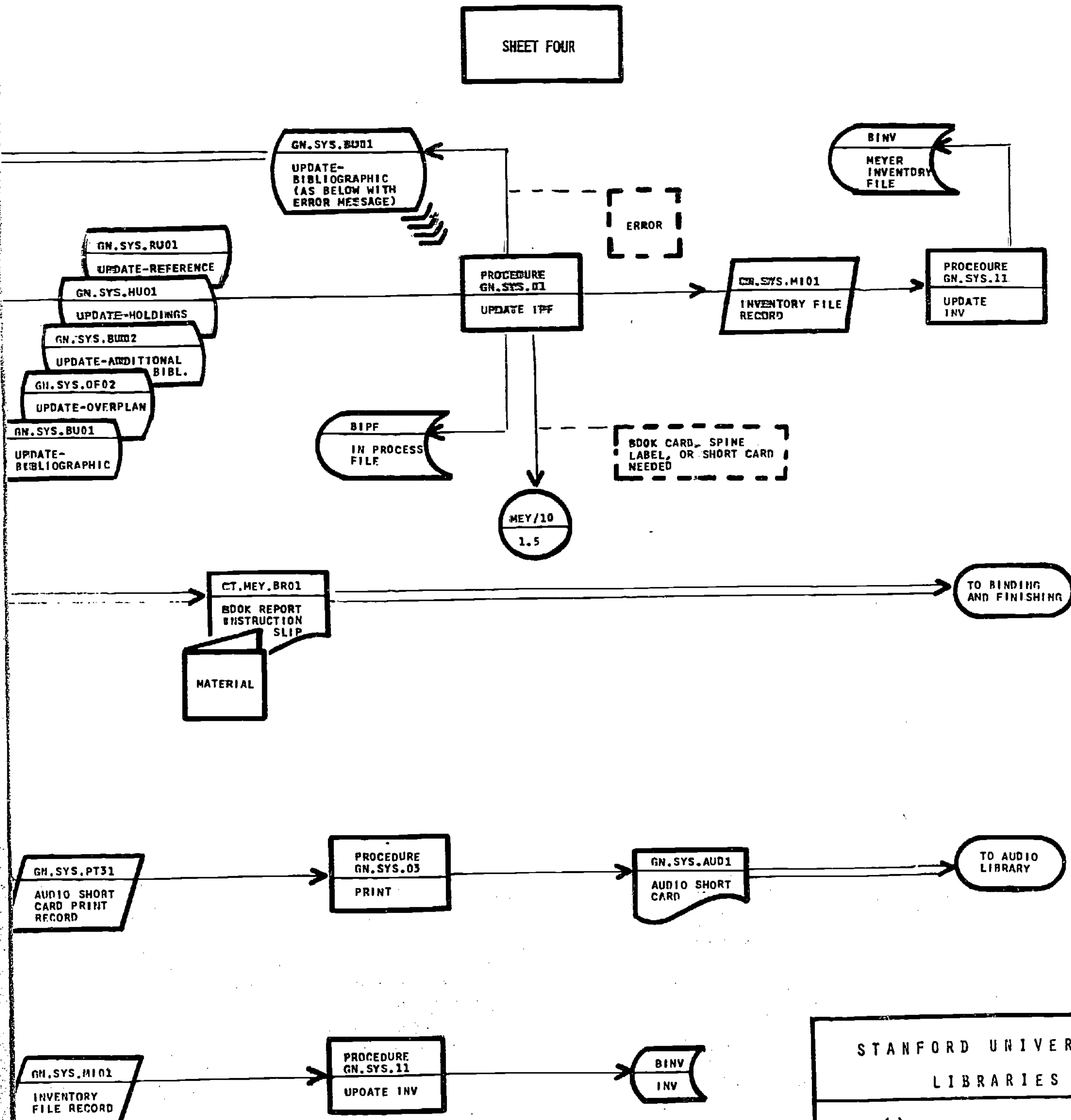

MATERIAL MAY ACCOMPANY REQUEST
UPDATES FROM ALL SOURCES
GN.SYS.S101
SEARCH INQUIRY (WITH ERROR MESSAGES)
FROM AUDIO LIBRARY
MATERIAL
PROCEDURE CT.HEY.02
KEY INV SEARCH
GN.SYS.S101
SEARCH INQUIRY
THE CALL NUMBER WILL ALREADY HAVE BEEN DETERMINED AND ATTACHED BY THE AUDIO LIB.

SHEET TWO


GN.SYS.S101
SEARCH INQUIRY (WITH ERROR MESSAGES)
BINV
MEYER INVENTORY FILE
DATA ELEMENT ERROR
PROCEDURE CT.MEY.02
KEY INV SEARCH
GN.SYS.S101
SEARCH INQUIRY
PROCEDURE GN.SYS.02
SEARCH
AT THE PRESENT TIME MARC DOES NOT CONTAIN RECORDS FOR PHONODISCS OR TAPES. WHEN THAT EXPANSION OCCURS, IT WILL BE POSSIBLE TO SEARCH MARC AND TRANSFER A RECORD TO THE IPF, UPDATE IT THERE, AND COPY IT TO THE INV.
MEY/11
3.4
BINV
MEYER INVENTORY FILE
ERROR
MEY/05
3.3
PROCEDURE GN.SYS.02
SEARCH
GN.SYS.S101
SEARCH INQUIRY (NO RECORDS FOUND)
GN.SYS.FH01
SEARCH RESULTS-FULL BIBL. AND PARTIAL HOLDINGS
PROCEDURE CT.MEY.03
MATCH REQUEST AGAINST SEARCH RESULTS
PROCEDURE CT.MEY.14
AUDIO CATALOGING
CONTINUE TO SEARCH FILE
ADDED COPY
MATERIAL
TO AUDIO LIBRARY
MUSIC TITLE II FILE
LC PRINTED MUSIC CATALOG



SHEET THREE
STOP
NO CATALOG RECORD MAINTENANCE REQUIRED
GN.SYS.SI01
SEARCH INQUIRY (NO RECORDS FOUND)
GN.SYS.PH01
SEARCH RESULTS-PARTIAL BIBL. AND FULL HOLDINGS
GN.SYS.FH01
SEARCH RESULTS-FULL BIBL. AND PARTIAL HOLDINGS
PROCEDURE CT.MEY.03
MATCH REQUEST AGAINST SEARCH RESULTS
PROCEDURE CT.MEY.19
REQUEST TRANSFER FROM INV TO IPF; KEY UPDATE; MOVE TO, OR DELETE FROM, INV
CATALOG RECORD MAINTENANCE NECESSARY
CHANGE ON MATERIAL REQUIRED
PROCEDURE CT.MEY.10
RECALL AND/OR REWORK MATERIAL
GN.SYS.BI01
INPUT-BIBLIOGRAPHIC (AS BELOW WITH ERROR MESSAGES)
DATA ELEMENT ERROR
ERROR, OR POSSIBLE ERROR
PROCEDURE CT.MEY.18
REVISION
PROCEDURE CT.MEY.16
KEY IPF RECORD; MOVE TO INV
GN.SYS.RI01
INPUT-REFERENCE
GN.SYS.BI02
INPUT-ADDITIONAL BIBLIOGRAPHIC
GN.SYS.OP01
INPUT-OVERPLAN
GN.SYS.BI01
INPUT-BIBLIOGRAPHIC
PROCEDURE GN.SYS.01
UPDATE IPF
MATERIAL
BIPF
IN PROCESS FILE
TO AUDIO LIBRARY



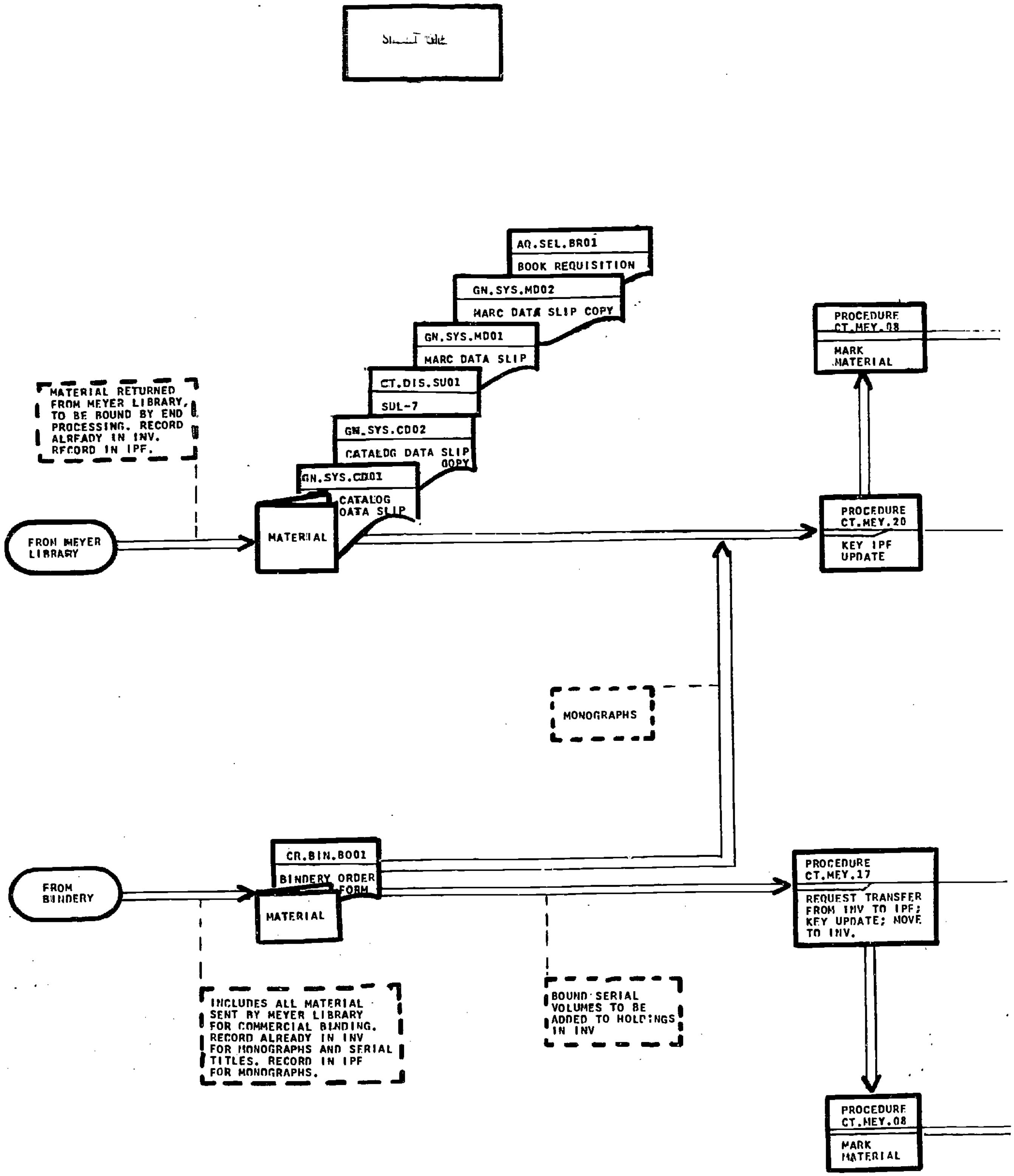
STANFORD UNIVERSITY LIBRARIES

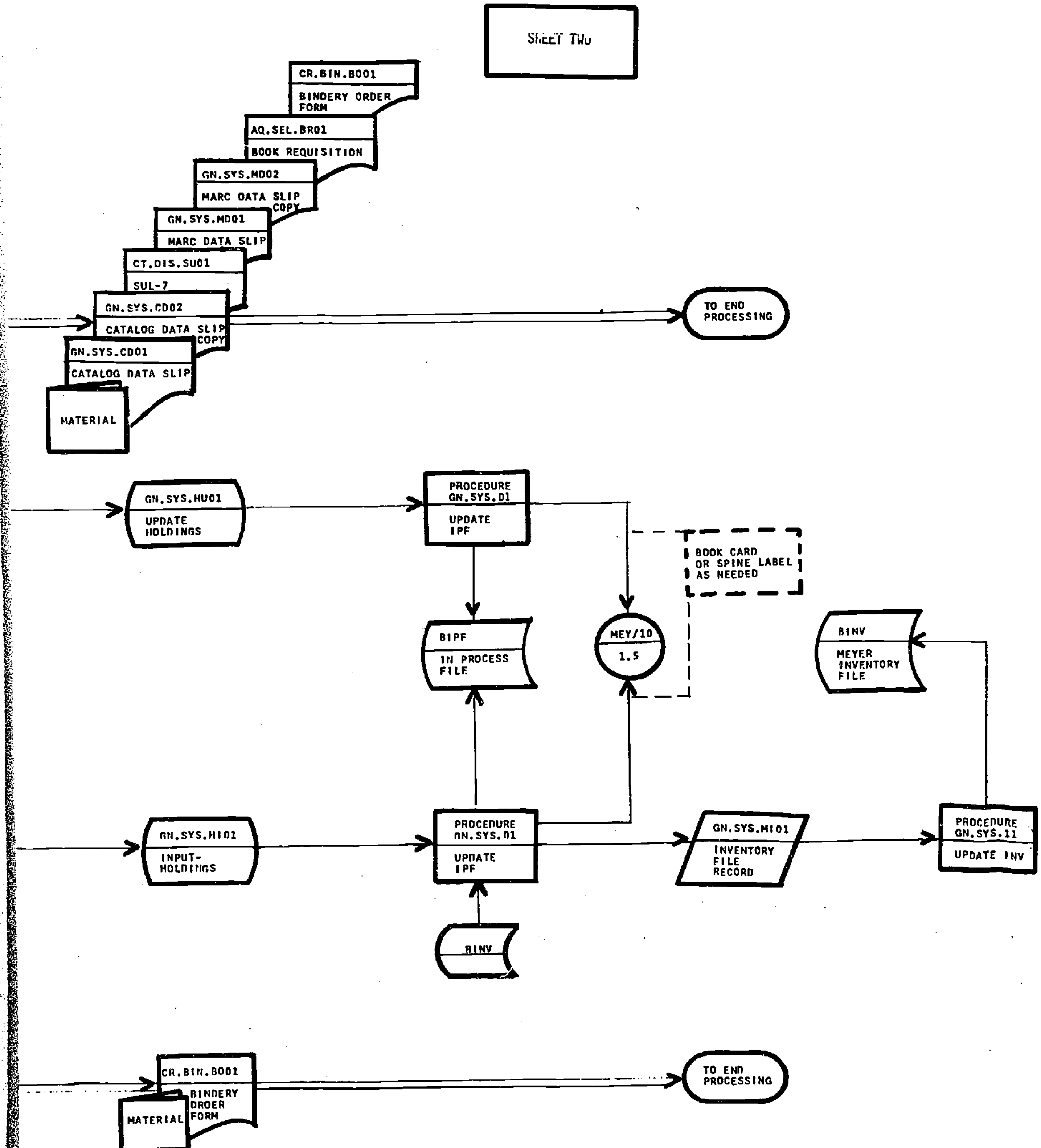
APPROVAL(S) REQUIRED: J. HITCHCOCK

REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEPT INIT
0	1/19	INITIAL VERSION	RP CK		

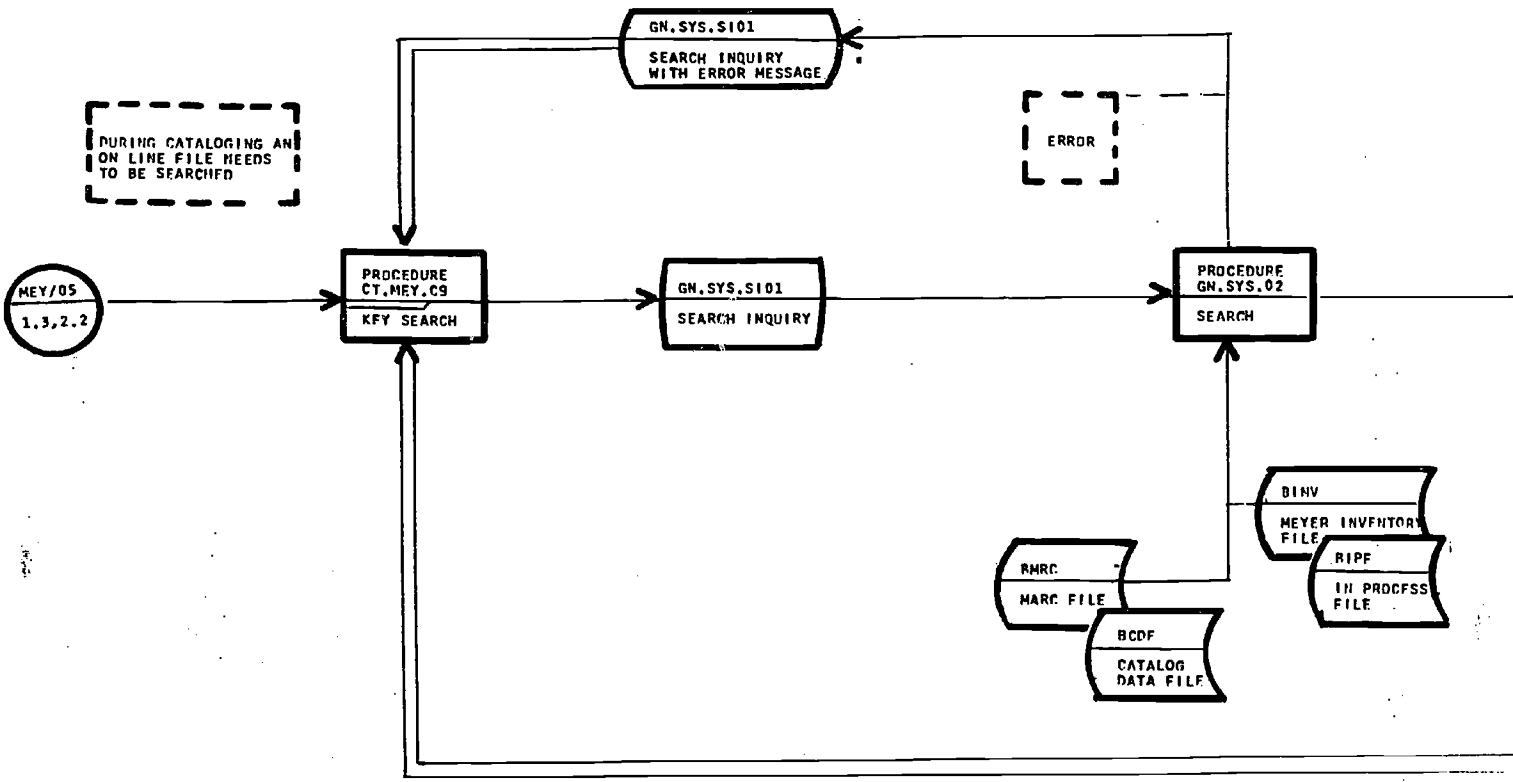
TITLE: MEYER CATALOGING

ID: CT.MEY


AQ.SEL.BR01
BOOK REQUISITION
GN.SYS.MD02
MARC DATA SLIP COPY
GN.SYS.MD01
MARC DATA SLIP
CT.DIS.SU01
SDL-7
GN.SYS.CD02
CATALOG DATA SLIP COPY
GN.SYS.CD01
CATALOG DATA SLIP
MATERIAL RETURNED FROM MEYER LIBRARY, TO BE BOUND BY END PROCESSING. RECORD ALREADY IN INV. RECORD IN IPF.
FROM MEYER LIBRARY
MATERIAL
PROCEDURE CT.MEY.08
MARK MATERIAL
PROCEDURE CT.MEY.20
KEY IPF UPDATE
MONOGRAPHS
CR.BIN.B001
BINDERY ORDER FORM
FROM BINDERY
MATERIAL
PROCEDURE CT.MEY.17
REQUEST TRANSFER FROM INV TO IPF; KEY UPDATE; MOVE TO INV.
INCLUDES ALL MATERIAL SENT BY MEYER LIBRARY FOR COMMERCIAL BINDING. RECORD ALREADY IN INV FOR MONOGRAPHS AND SERIAL TITLES. RECORD IN IPF FOR MONOGRAPHS.
BOUND SERIAL VOLUMES TO BE ADDED TO HOLDINGS IN INV
PROCEDURE CT.MEY.08
MARK MATERIAL

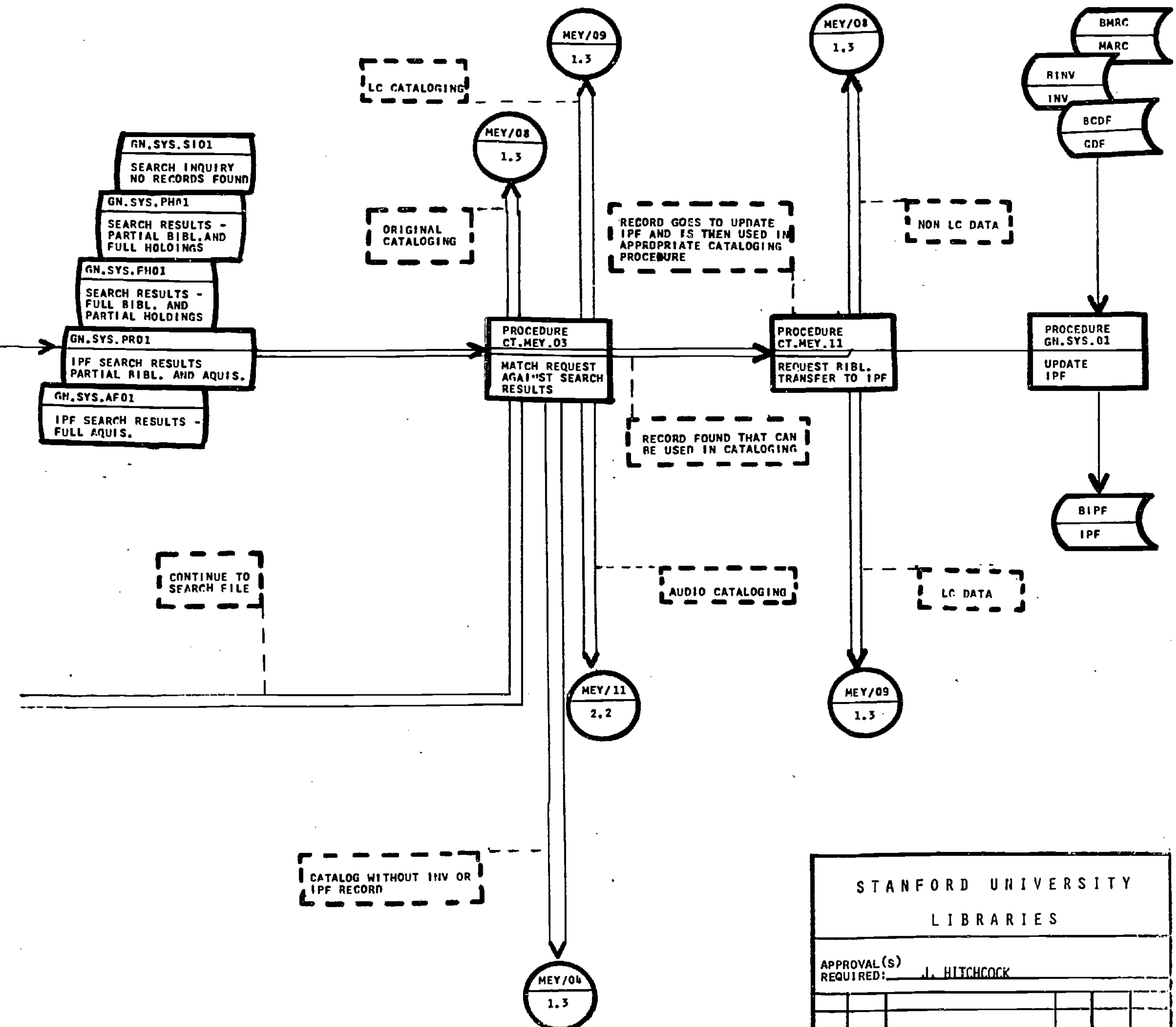

SHEET TWO
CR.BIN.B001
BINDERY ORDER FORM
AQ.SEL.BR01
BOOK REQUISITION
GN.SYS.MD02
MARC DATA SLIP COPY
GN.SYS.MD01
MARC DATA SLIP
CT.DIS.SU01
SUL-7
GN.SYS.CD02
CATALOG DATA SLIP COPY
GN.SYS.CD01
CATALOG DATA SLIP
MATERIAL
TO END PROCESSING
GN.SYS.HU01
UPDATE HOLDINGS
PROCEDURE GN.SYS.01
UPDATE IPF
BOOK CARD OR SPINE LABEL AS NEEDED
BIPF
IN PROCESS FILE
MEY/10
1.5
BINV
MEYER INVENTORY FILE
GN.SYS.HI01
INPUT-HOLDINGS
PROCEDURE GN.SYS.01
UPDATE IPF
GN.SYS.MI01
INVENTORY FILE RECORD
PROCEDURE GN.SYS.11
UPDATE INV
BINV
CR.BIN.B001
BINDERY ORDER FORM
MATERIAL
TO END PROCESSING

SHEET THREE



MEY/05
1.3.2.2
DURING CATALOGING AN ON LINE FILE NEEDS TO BE SEARCHED
PROCEDURE CT.MEY.C9
KEY SEARCH
GN.SYS.S101
SEARCH INQUIRY WITH ERROR MESSAGE
GN.SYS.S101
SEARCH INQUIRY
ERROR
PROCEDURE GN.SYS.02
SEARCH
BINV
MEYER INVENTORY FILE
BIPF
IN PROCESS FILE
BMRC
MARC FILE
BCDF
CATALOG DATA FILE

SHEET FOUR



STANFORD UNIVERSITY LIBRARIES

APPROVAL(S) REQUIRED: J. HITCHCOCK

REV NO.	REV DATE	REVISION REMARKS	ANAL INIT	ASST INIT	DEPT INIT
0	1/19	INITIAL VERSION	RP CK		

TITLE: MEYER CATALOGING

CT.MEY

OUTPUT RECORD FORMAT	SUBSYSTEM: GN	DATE: 9/11/71
	PROCESS: SYS	REVISED:
	PROCEDURE: 03	AUTHOR: CK
	FORM NO. SUL-200	

PAGE 1 OF 1

1

2

3 (a) NAME: Abel Original Invoice — (b) ID NUMBER: GN.SYS.AI01

4 (a) SIZE: 3 1/2" x 7 7/8" (8.9 cm x 20cm) — (b) PART 1 of 2

5 FORMAT:

STANFORD UNIVERSITY LIBRARIES

ABEL ORIGINAL INVOICE

DATE OF ORDER

ORDER NO.

NO. COPIES

TOTAL EST. PRICE ▶

VEND. NO.

BILL IN DUPLICATE TO: ORDER DEPT. - STANFORD UNIVERSITY LIBRARIES STANFORD, CALIFORNIA 94305

SUL-200 (1-69)

DEALER LEAVE BLANK ▶	INV DATE	L.P.	N.P.	S. TX.	S.P.

DEALER: SEE OTHER SIDE

F-5-(8/70)

OUTPUT RECORD CONTENT	1. NAME: Abel Original Invoice	DATE: 9/11/70	REVISED: 5/17/71	PAGE 1 OF 1
	2. ID NUMBER: GN.SYS.AI01	AUTHOR: GC		

3. DATA ELEMENT	4. SOURCE	5. OCCURRENCE Min.	Max.	Prob.	6. EDIT RULES	7. STORAGE RULES
Please see B-20 for GN.SYS.P00						

B-20 (Rev. 9/70)

PROCESSING RULES	PROCESS: SYS	REVISED:	OF 1
	PROCEDURE:	AUTHOR: EAM	

2 3

RULE NO.	RULE TEXT
***GN.SYS.AI01	
	See processing rules for GN.SYS.P001 (Purchase Order)

unlined B-7 (Rev. 9/70)

SYSTEM BOOK #2

1 OUTPUT RECORD FORMAT	SUBSYSTEM: GN	DATE: 9/11/70	PAGE 1 OF 1
	PROCESS: SYS	REVISED:	
	PROCEDURE: 03	AUTHOR: CK	
2	FORM NO. SUL-200		

3 (a) NAME: Abel Accounts Receivable Copy | (b) ID NUMBER: GN.SYS.AS01

4 (a) SIZE: 3 1/2" x 7 7/8" (8.9 cm x 20 cm) | (b) PART 1 of 1

5 FORMAT:

STANFORD UNIVERSITY LIBRARIES [redacted] DATE OF ORDER | ORDER NO.

NO. COPIES

TOTAL EST. PRICE ▶ | VEND. NO. | BILL IN DUPLICATE TO: ORDER DEPT. - STANFORD UNIVERSITY LIBRARIES STANFORD, CALIFORNIA 94305

SUL-200 (1-69)

DEALER LEAVE BLANK ▶	INV. DATE	L.P.	N.P.	S. TX.	S.P.	DATE RECD.

DEALER: SEE OTHER SIDE

B-5a (8/70)

OUTPUT RECORD CONTENT	1. NAME: Abel Accounts Receivable Copy	DATE: 9/11/70	REVISED: 5/17/71	PAGE 1 OF 1
	2. ID NUMBER: GN.SYS.AS01	AUTHOR: GC		

3. DATA ELEMENT	4. SOURCE	5. OCCURRENCE Min.	Max.	Prob.	6. EDIT RULES	7. STORAGE RULES
Please see B-20 for GN.SYS.POO1.						

B-20 (Rev. 9/70)

S3

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 9/17/71	PAGE 1 OF 1
	PROCESS: SYS	REVISED:	
	PROCEDURE:	AUTHOR: EAM	

2 RULE NO.	3 RULE TEXT

***GN.SYS.AS01

See processing rules for GN.SYS.PO01 (Purchase Order)

1 OUTPUT RECORD FORMAT	SUBSYSTEM: GN	DATE: 9/12/70	PAGE 1 OF 2
	PROCESS: SYS	REVISED:	
	PROCEDURE: 03	AUTHOR: GC	
2	FORM NO. SUL-200		

3 (a) NAME: Cover Sheet (b) ID NUMBER: GN.SYS.P001

4 (a) SIZE: 3 1/2" x 7 7/8" (b) PART 1 of 2

5 FORMAT:

STANFORD UNIVERSITY LIBRARIES [redacted] DATE OF ORDER ORDER NO.

NO. COPIES

1 Richard Abel & Co., Inc.
2 Industrial Center Bldg.
3 Marinship Gate 5 Road
4 Sausalito, CA 94965

TOTAL EST. PRICE ▶ VEND. NO. BILL IN DUPLICATE TO: ORDER DEPT. · STANFORD UNIVERSITY LIBRARIES STANFORD, CALIFORNIA 94305

SUL-200 (1-69)

DEALER LEAVE BLANK ▶	INV. DATE	L.P.	N.P.	S. TX.	S.P

DEALER: SEE OTHER SIDE

180

1 OUTPUT RECORD FORMAT	SUBSYSTEM: GN	DATE: 9/11/70
	PROCESS: SYS	REVISED:
	PROCEDURE: 03	AUTHOR: GC
2	FORM NO. SUL-200	

3 (a) NAME: Purchase Order	(b) ID NUMBER: GN.SYS.P001
4 (a) SIZE: 3 1/2" x 7 3/8"	(b) PART 1 of 2

5 FORMAT

99132

STANFORD UNIVERSITY LIBRARIES

#1 PURCHASE ORDER #2 DATE OF ORDER 4-29-69 #3 ORDER NO. 69-2581

[redacted]

NO. COPIES 1 #8

Tarski, Alfred, 1902-
Undecidable theories. In collaboration with Andrzej Mostowski
and Raphael M. Robinson. 2d printing ed. Amsterdam, North-Holland,
1968.

(Studies in logic and foundations of mathematics.)

#4

SHIP AND BILL IN DUPLICATE TO:
Order Department
Stanford University Libraries
Stanford, CA 94305
#5

TOTAL EST PRICE ▶ F1 15. #6 VEND. NO. ABEL #7

SHIP & BILL IN DUPLICATE TO: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANFORD, CALIFORNIA 94305

SUL-200 (1-69)

DEALER: SEE OTHER SIDE

DEALER LEAVE BLANK ▶ INV. DATE | L.P. | N.P. | S. TX. | S.P.

1 3 4 5 12 18

NCR paper

UARCO BUSINESS FORMS OAKLAND, CALIF.

B-5a (8/70)

SYSTEM BOOK #2

INPUT RECORD CONTENT	1. NAME: Purchase Order Print Record	DATE: 9/12/70	REVISED: 3/2/71	PAGE 1
	2. ID NUMBER: GN.SYS.PT01	AUTHOR: GC		OF 1

3. DATA ELEMENT	4. SOURCE	5. OCCURRENCE Min.	Max.	6. Prob.	EDIT RULES	7. STORAGE RULES
CRD	BMRC	1	1	1		
ADD	OR01	1	1	1		
PR	OR01	0	1	1		
VI or { VN / VSD / VCS }	OR01	1	1	1		
ORD	machine generated	1	1	1		
PO	OR01	0	1	.3		
DAN	OR01	0	1	.02		
VCT	OR01	0	1	.1		
VSP	OR01	0	1	.4		
ME-PN	BMRC	1	1	1		
-CA / -CF / -T / -UT } one of these						
TS-T	BMRC	0	1	.9		
-SB	BMRC	0	1	.5		
-RT	BMRC	0	1	.6		
-UT	BMRC	0	1	.2		
ED	BMRC	0	1	.8		
PP	BMRC	1	1	1		
D	BMRC	1	1	1		
PUX	BI01	0	1	.3		
RIP	BI01	0	1	.2		
SN-PN } one	BMRC	0	1	.4		
-CA } of						
-CF } these	BMRC					
-ST	BMRC	0	1	.5		
-SV	BMRC	0	1	.5		
-SP	BI02	0	1	.2		
MDX	OR01	1	1	1		

182

B-9 (Rev. 10/70)

SYSTEM BOOK #2

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 6/23/71	PAGE 1
	PROCESS: SYS	REVISED:	OF 1
	PROCEDURE: 03	AUTHOR: LK/GC	

RULE NO. (2)	RULE TEXT (3)
****GN.SYS.PT01	See processing rules for GN.SYS.P001 (Purchase Order).

183

SYSTEM BOOK #2

OUTPUT RECORD CONTENT	1. NAME: Purchase Order	DATE: 9/11/70	REVISED: 4/12/71	PAGE 1 OF 1
	2. ID NUMBER: GN.SYS.P001	AUTHOR: LK		

3 DATA ELEMENT	4 SOURCE	5 OCCURRENCE Min.	Max.	6 Prob.	DISPLAY RULES	7 RETRIEVAL RULES
CRD	GN.SYS.PT01 ↓	1	1	1		
ADD		1	1	1		
PR		0	1	1		
VI or VN		1	1	1		
VSD						
VCS						
ORD		1	1	1		
PO		0	1	.9		
DAN		0	1	.1		
VCT		0	1	.2		
VSP		0	1	.4		
ME-PN }		1	1	1		
-CA }						
-CF }						
-T }						
-UT }						
TS-T		0	1	.9		
-SB		0	1	.7		
-RT		0	1	.6		
-UT		0	1	.2		
ED		0	1	.9		
PP		1	1	1		
D		1	1	1		
PUX		0	1	.3		
RIP		0	1	.2		
SN-PN } one		0	1	.4		
-CA } of						
-CF } these						
-ST		0	1	.5		
-SV		0	1	.5		
-SP		0	1	.2		
MDX		1	1	1		

B-20 (Rev. 9/70)

SYSTEM BOOK #2

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 8/2/71	PAGE 1
	PROCESS: SYS	REVISED:	OF 7
	PROCEDURE: 03	AUTHOR: GC/LK	

2 3

RULE NO. **RULE TEXT**

****GN.SYS.P001

107.00 35000
Sort by value of VI or VN, whichever is present.

107.01 24000.39
All lines will be terminated at nearest whole word or at the hyphen of a hyphenated word. Never break before a hyphen. Never break between two hyphens.

107.02 24000.58
Count the number of records entering the program .
Count the number of purchase orders printed.
Count the number of records not processed.
Count the number of Abel forms printed.

107.03 24000.01
For Formatting of GN.SYS.P001 use form SUL200, lines 1-18, columns 2-7

107.04 24000.04
Cover sheet
At the beginning of each vendor change, a blank form will be used as an address sheet.
If VI is present in the record, get the vendor name and address from the vendor address table.
If VI is not present, obtain the vendor name and address from the data elements VN and VSD and VCS in the record.
If Vendor Name is obtained from table, print the name and address as stored starting in line 11, column 7.
The Vendor Name is allowed 31 spaces on each of two lines.
If Vendor Name and Address is obtained from record and VN has more than 31 characters, start in line 11, column 7. Continue printing VN on line 12 column 7. Break the two lines at a space or hyphen.
If VN has less than 31 characters, start in line 12, column 7.
Print VSD in line 13, column 7.
Print VCS in line 14, column 7.
Print a new cover sheet after every 32 forms for a single vendor. If VI=ABEL print a new cover sheet after every 16 purchase orders and 16 Abel original invoices (32 forms).

107.05 24000.02
Title of form: Area #1
On line 1, column 21 print
'P U R C H A S E   O R D E R'
On line 3, column 17 print
'RETURN WITH MATERIAL OR USE AS REPORT'

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 8/2/71	PAGE 2
	PROCESS: SYS	REVISED:	OF 7
	PROCEDURE: 03	AUTHOR: LK/GC	

2 RULE NO.	3 RULE TEXT

107.06 24000.03
If VI=ABEL, print a second copy of the SUL-200. For these copies
On line 1, column 24 print
'ABEL ORIGINAL INVOICE'
On line 3, column 20 print
'ABEL ACCOUNTS RECEIVABLE COPY'

107.07 24000.05
Date of order: Area #2
'3,55'::=<ORD>

107.08 24000.06
Order number: Area #3
'3,65'::=<CRD>

107.09 24000.07
If ADD=01,02,05,06,07, or 15
'12,44'::=SHIP<SP(1)>AND<SP(1)>BILL<SP(1)>IN
<SP(1)>DUPLICATE<SP(1)>TO:
'13,44'::=<address from Ship to Address table
pointed to by value of ADD>
'14,44'
'15,44'
'16,44'
'18,44'::=X(30)

If ADD=03,04,08,09,10,11,12,13, or 14
'13,44'::=<address from Ship to Address table
pointed to by value of ADD>
'14,44'
'15,44'
'16,44'

Note: The values of data element ADD are codes which direct the program to the Ship to Address table. The table contains preformatted ship to addresses for all ship to addresses.

107.10 24000.08
Total estimated price: Area #6
'18,10'::=(<PR>)

107.11 24000.09
Vendor number: Area #7
'18,36'::=(<VI>)
|<1st 7 non-blank, non-special characters of VN
with blanks and special characters compressed out.>

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 9/2/71	PAGE 3
	PROCESS: SYS	REVISED: 12/20/71	OF 7
	PROCEDURE: 03	AUTHOR: LK/GC	

2 3

RULE NO. **RULE TEXT**

```
107.12   24000.10
  Number of copies:  Area #8
  Print the value of MDX until the first delimiter (a left paren)
  in line 5, column 2.  Every purchase order must have a number
  of copies.  Do not print the left paren.  Do not print <SP(1)>C
  or <SP(1)>C. or <SP(1)>C, etc.  The portion of MDX delimited by
  left paren is optional.

107.13   24000.11
  Bibliographic information:  Area #4
  If VSP contains 'RUSH', 'rush', or 'Rush' then
    '4,6'::=<*(28)>PLEASE RUSH<*(28)>

107.14   24000.12
  If MET is present and if VSP contains 'RUSH', 'Rush', or 'rush'
  then '5,7'::=<MET>(SP(1)><TSSB>)(<SP(1)><TSRT>)(<SP(2)><ED>)<SP(2)>
               <PP><SP(1)><D>(<SP(2)><PUX>)
  Else if MET is present
  '4,7'::=<MET>(<SP(1)><TSSB>)(<SP(1)><TSRT>)(<SP(2)><ED>)<SP(2)>
          <PP><SP(1)><D>(SP(2)><PUX>)

  Each succeeding line will begin in column 11.

107.15   24000.13
  If MET is not present and if VSP contains RUSH, Rush or rush then
  '5,7'::=(<MEPN>)
         |(<MECA>)
         |(<MECF>)
         |(<MEUT>)
  Else if MET is not present
  '4,7'::=(<MEPN>)
         |(<MECA>)
         |(<MECF>)
         |(<MEUT>)

  Each succeeding line will begin in column 11.

107.16   24000.17
  'L+1,11'::=(<TSUT>)

  Each succeeding line will begin in column 11.
```

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 8/2/71	PAGE 4
	PROCESS: SYS	REVISED:	OF 7
	PROCEDURE: 03	AUTHOR: LK/GC	

2 RULE NO.	3 RULE TEXT

107.17 24000.18

If MET is not present

```
'L+1,11'::=<TST>(<SP(1)><TSSB>)(<SP(1)><TSRT>)(<SP(2)>
          <ED>)<SP(2)><PP><SP(1)><D>(<SP(2)><PUX>)
```

Each succeeding line will begin in column 2.

107.18 24000.19

If MET is present, TST will not be printed.

107.19 24000.23

If RIP is present, print its value in place of PP.
Else print the value of PP.

107.20 24000.25

Series Notes

```
'L+1,11'::=('('<SNPN><SP(2)><SNST><SP(1)><SNSV>')')
          |('('<SNPN><SP(2)><SNST>')')
          |('('<SNCA><SP(2)><SNST><SP(1)><SNSV>')')
          |('('<SNCA><SP(2)><SNST>')')
          |('('<SNCF><SP(2)><SNST><SP(1)><SNSV>')')
          |('('<SNCF><SP(2)><SNST>')')
          |('('<SNST><SP(1)><SNSV>')')
          |('('<SNST>')')
```

Each succeeding line will begin in column 2.

107.21 24000.26

Only the first occurrence of series notes will be printed.

107.22 24000.30

```
'L,C+2'::=('('<SNSP>')')
```

Each succeeding line will begin in column 2.

107.23 24000.36

Order or Subscription Information.
Only the optional portion of MDX, that delimited by parentheses in the input, is used here. The required part of MDX (the number of copies part) is used in 107.12. Print the optional part of MDX as input as described below.

If PO=1

```
'L+1,2'::=SUBSCRIPTION<SP(1)>TO<SP(1)>BEGIN<SP(1)>WITH
          <SP(1)><MDX><SP(1)>AND<SP(1)>TO<SP(1)>CONTINUE
          <SP(1)>UNTIL<SP(1)>FURTHER<SP(1)>NOTICE.
```

1			
PROCESSING RULES	SUBSYSTEM: GN	DATE: 8/2/71	PAGE 5
	PROCESS: SYS	REVISED:	OF 7
	PROCEDURE: 03	AUTHOR: LK/GC	

2 RULE NO.	3 RULE TEXT

```
     IF PO=2
     'L+1,2'::=<MDX><SP(1)>AND<SP(1)>ALL<SP(1)>FUTURE
               <SP(1)>VOLUMES<SP(1)>AS<SP(1)>PUBLISHED.

     IF PO=3
     'L+1,2'::=(<MDX><SP(1)>)ALL<SP(1)>VOLUMES<SP(1)>
               PUBLISHED<SP(1)>AND<SP(1)>A<SP(1)>STANDING
               <SP(1)>ORDER<SP(1)>FOR<SP(1)>FURTHER<SP(1)>VOLUMES.

     IF PO=4
     'L+1,2'::=(<MDX><SP(2)>)DO<SP(1)>NOT<SP(1)>DUPLICATE
               <SP(1)>ON<SP(1)>BLANKET<SP(1)>ORDER.

     IF PO=6
     'L+1,2'::=(<MDX><SP(2)))>)DO<SP(1)>NOT<SP(1)>DUPLICATE
               <SP(1)>ON<SP(1)>STANDING<SP(1)>ORDER.

     IF PO=7
     'L+1,2'::=(<MDX><SP(2)>)PLEASE<SP(1)>QUOTE<SP(1)>ON
               <SP(1)>BACK<SP(1)>ISSUES.

     IF PO=8
     'L+1,2'::=(<MDX><SP(1)>)PLEASE<SP(1)>CHARGE<SP(1)TO
               <SP(1)>STANFORD<SP(1)>UNIVERSITY<SP(1)>
               LIBRARIES<SP(1)>DEPOSIT<SP(1)>ACCOUNT:
               <SP(1)><DAN>

     IF PO=9
     'L+1,2'::=(<MDX><SP(2)>)DO<SP(1)>NOT<SP(1)>
               DUPLICATE<SP(1)>ON<SP(1)>U.P.S.O.

     PO Messages 4, 6, 7 and 9 may be used in conjunction with
     PO messages 1, 2, 3.
     If more than one code is present in the data element PO,
     print according to the rule for the first PO code present and print t
     following PO messages on the same line 2 spaces following
     the preceeding message, using the rule for the second PO
     code, etc.  DO NOT REPRINT MDX in printing subsequent PO MESSAGES

  If PO is not present
  'L+1,2'::=(<MDX>)

  107.24     24000.37
   'L+1,2'::=(<VSP>)
```

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 8/2/71	PAGE 6
	PROCESS: SYS	REVISED:	OF 7
	PROCEDURE: 03	AUTHOR: LK/GC	

RULE NO. 2	RULE TEXT 3

Capitalize and underline the value of VSP.

107.25 24000.38
Vendor catalog information
'L+1,2'::=(<VCT>)

107.26 24000.40
Overflow and error logic
Overflow on to a second form
Will occur if the length of the data elements values to be printed in Area #4 is greater than 794 characters.

107.27 24000.41
If overflow occurs
'17,2'::=CONTINUED<SP(1)>ON<SP(1)>NEXT<SP(1)>PAGE.
On the first form

107.28 24000.45
If overflow occurs, print on the second form
'3,65'::=<CRD>
'1,26'::=P A G E T W O
'3,26'::=P A G E T W O

and continue printing Area #4
in line 4 column 7. Line 5 will begin in column 7.

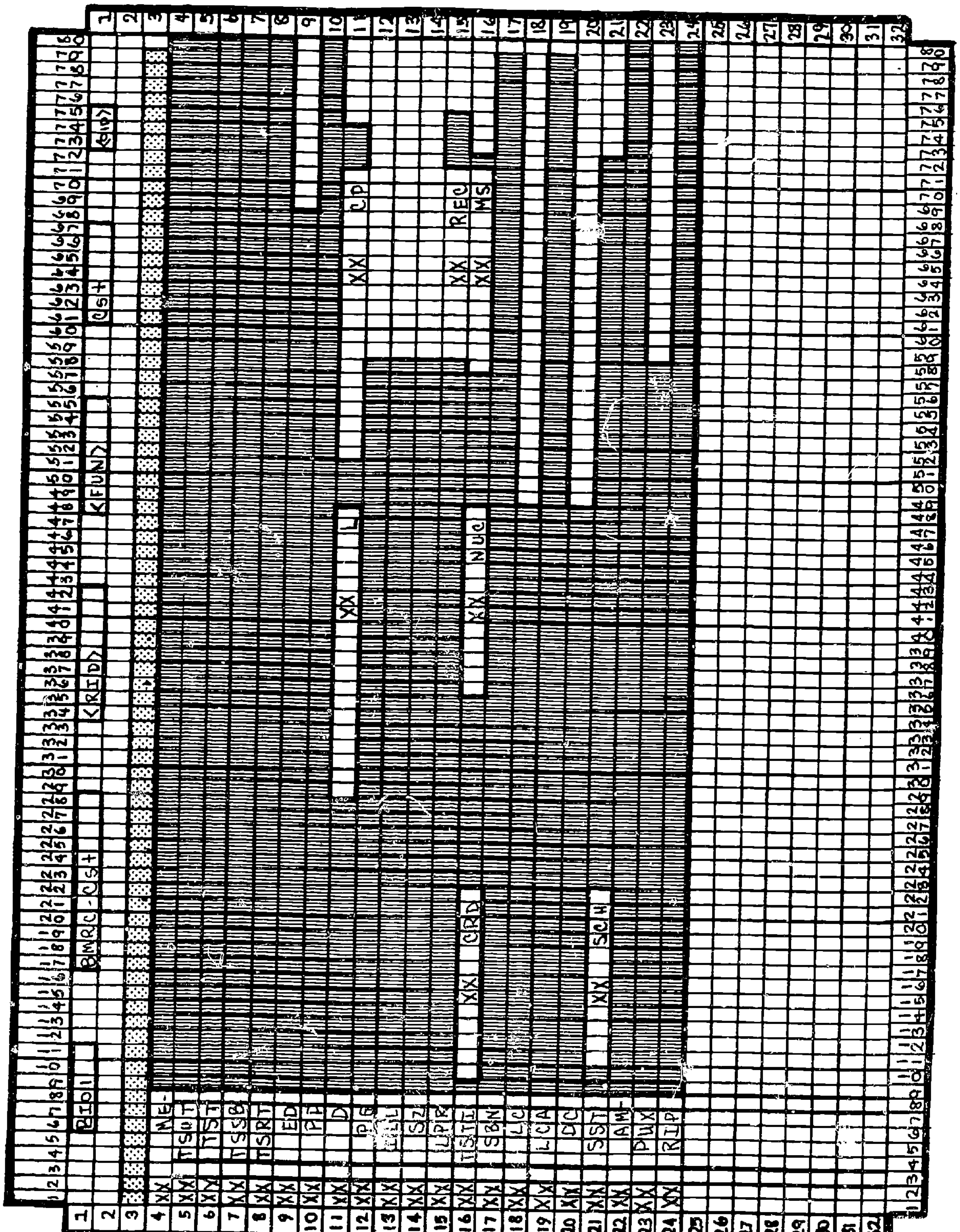
107.29 24000.46
Do not allow overflow onto a third purchase order form.
If length of Area #4 exceeds 1588 characters, write the formatted record at the end of the print run with the message:
CARD NO. <CRD>> not processed. Record length too long.
on line 1 of 1st page. Overflow as many times as necessary.
The file manager will review the record for errors with the Order Division. If VI=ABEL in this error record do not print twice.

107.30 24000.48
End of Job Summary report
On end of run put page
'1,25'::=END<SP(1)>OF<SP(1)>JOB<SP(1)>SUMMARY

1 PROCESSING RULES	SUBSYSTEM: GN	DATE: 8/2/71	PAGE 7
	PROCESS: SYS	REVISED:	
	PROCEDURE: 03	AUTHOR: LK/GC	OF 7

2 RULE NO.	3 RULE TEXT
107.31	24000.49 '3,56'::=<DATE RETURNED BY DATE FUNCTION,IN FORMAT MM/DD/YY>
107.32	24000.50 '4,7'::=RECORDS<SP(1)>ENTERING<SP(1)>PROGRAM
107.33	24000.51 '4,47'::=PURCHASE<SP(1)>ORDERS<SP(1)>PRINTED
107.34	24000.52 In line 5, column 17 print the value of number of records passed to print program.
107.35	24000.53 In line 5, column 57 print the value of number of purchase orders printed.
107.36	24000.54 '7,7'::=NUMBER<SP(1)>NOT<SP(1)>PROCESSED
107.37	24000.55 '7,47'::=ABEL<SP(1)>FORMS<SP(1)>PRINTED
107.38	24000.56 In line 8, column 17 print the value of records not processed.
107.39	24000.57 In line 8, column 57 print the value of number of purchase orders printed twice.
107.40	24000.59 If any record can not be processed, print an identifying value and a descriptive error diagnostic on the page following the end of job summary.

CRT FORMAT	ID: BI01 (GN.SYS.BI01)	DATE: 1/20/72	PAGE 1 OF 1
	NAME: Input-Bibliographic	REVISED:	
	TITLE:	AUTHOR: WED	



APPENDIX C

SAMPLES OF DATA ELEMENT STATISTICAL ANALYSIS BASED ON MARC TAPES

**** STATISTICS FOR DATA ELEMENT MEPN* ****

MARC TAPE 08-13-70

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
6	1	1	0.29
[illegible]	1	2	0.58
10	1	3	0.87
11	4	7	2.03
12	9	16	4.65
13	9	25	7.27
14	18	43	12.50
15	33	76	22.09
16	14	90	26.16
17	24	114	33.13
18	16	130	37.79
19	24	154	44.77
20	20	174	50.58
21	21	195	56.68
22	13	208	60.46
23	19	227	65.98
24	10	237	68.89
25	6	243	70.63
26	17	260	75.58
27	15	275	79.94
28	12	287	83.43
29	9	296	86.04
30	7	303	88.08
31	11	314	91.27
32	5	319	92.73
33	1	320	93.02
34	5	325	94.48
35	3	328	95.34
36	4	332	96.51
37	3	335	97.38
38	3	338	98.25
40	1	339	98.54
41	2	341	99.13
49	1	342	99.41
53	2	344	100.00

TOTALS FOR FILE:

NUMBER ENTRIES EXAMINED = 500

NUMBER DATA ELEMENTS FOUND = 344

NUMBER ENTRIES WITH NO OCCURRENCE OF DATA ELEMENT = 156

**** S T A T I S T I C S   F O R   D A T A   E L E M E N T   MEPN* ****

MARC T A P E 08-13-70 PAGE 2

****

NUMBER OCCURRENCES OF MULTIPLE DATA ELEMENTS = 0

****

STATISTICS CALCULATED:

MIN = 6

MEAN = 21.74

MAX = 53

VARIANCE = 52.04

STANDARD DEVIATION = 7.21

SKEW = 1.02

KURTOSIS = 1.64

*Main Entry-Personal Name

**** STATISTICS FOR DATA ELEMENT MEPN* ****

MARC TAPE 10-01-70

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
10	2	2	0.48
11	2	4	0.95
12	10	14	3.35
13	15	29	6.95
14	18	47	11.27
15	17	64	15.34
16	25	89	21.34
17	20	109	26.13
18	23	132	31.65
19	16	148	35.48
20	21	169	40.52
21	21	190	45.56
22	21	211	50.59
23	21	232	55.63
24	16	248	59.47
25	18	266	63.78
26	17	283	67.86
27	19	302	72.42
28	12	314	75.30
29	11	325	77.94
30	19	344	82.49
31	15	359	86.09
32	9	368	88.24
33	11	379	90.88
34	6	385	92.32
35	8	393	94.24
36	6	399	95.68
37	5	404	96.88
38	4	408	97.84
39	1	409	98.08
40	2	411	98.55
42	1	412	98.80
44	2	414	99.27
45	1	415	99.52
50	1	416	[illegible].76
52	1	417	[illegible]0.00

TOTALS FOR FILE:

NUMBER ENTRIES EXAMINED = 500

NUMBER DATA ELEMENTS FOUND = 417

*Main Entry-Personal Name

**** S T A T I S T I C S   F O R   D A T A   E L E M E N T MEPN* ****

MARC T A P E 10-01-70   PAGE 2

****

NUMBER ENTRIES WITH NO OCCURRENCE OF DATA ELEMENT =   83

NUMBER OCCURRENCES OF MULTIPLE DATA ELEMENTS =   0

****

STATISTICS CALCULATED:

MIN =   10

MEAN =   23.23

MAX =   52

VARIANCE =   54.79

STANDARD DEVIATION =   7.40

SKEW =   0.61

KURTOSIS =   0.16

*Main Entry-Personal Name

**** S T A T I S T I C S   F O R   D A T A   E L E M E N T NB* ****

MARC T A P E 10-01-70   PAGE 1

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
19	1	1	0.40
20	12	13	5.23
21	11	24	9.67
22	3	27	10.88
23	20	47	18.95
24	31	78	31.45
25	60	138	55.64
26	3	141	56.85
27	19	160	64.52
28	1	161	64.91
29	1	162	65.32
33	3	165	66.53
35	2	167	67.34
36	80	247	99.59
149	1	248	100.00

****

TOTALS FOR FILE:

NUMBER ENTRIES EXAMINED = 500

NUMBER DATA ELEMENTS FOUND = 248

NUMBER ENTRIES WITH NO OCCURRENCE OF DATA ELEMENT = 253

NUMBER OCCURRENCES OF MULTIPLE DATA ELEMENTS = 1

MULTIPLICITY	NUMBER OCCURRENCES
2	1

****

STATISTICS CALCULATED:

MIN = 19

MEAN = 28.65

MAX = 149

VARIANCE = 92.68

STANDARD DEVIATION = 9.63

*Notes-Bibliography

****

SKEW = 7.96

KURTOSIS = 96.48

*Notes-Bibliography

**** S T A T I S T I C S F O R D A T A E L E M E N T OACA* ****

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
17	1	1	0.83
18	1	2	1.66
19	1	3	2.50
22	3	6	5.00
23	1	7	5.83
24	2	9	7.50
25	2	11	9.16
27	2	13	10.83
28	4	17	14.16
29	7	24	20.00
30	1	25	20.83
31	2	27	22.50
32	6	33	27.50
33	1	34	28.33
34	6	40	33.33
35	2	42	35.00
36	5	47	39.16
37	3	50	41.66
38	3	53	44.16
39	3	56	46.66
40	2	58	48.33
41	1	59	49.16
42	2	61	50.83
43	5	66	55.00
44	3	69	57.50
45	1	70	58.33
46	1	71	59.16
47	1	72	60.00
48	3	75	62.50
49	2	77	64.16
51	1	78	65.00
52	3	81	67.50
53	3	84	70.00
55	2	86	71.66
56	1	87	72.50
57	2	89	74.16
58	1	90	75.00
59	3	93	77.50
60	1	94	78.33
62	1	95	79.16
63	3	98	81.66
64	1	99	82.50
65	1	100	83.33
66	1	101	84.16
68	1	102	85.00

*Other Added Entries-Corporate Name

**** S T A T I S T I C S   F O R   D A T A   E L E M E N T OACA* ****

MARC T A P E 08-13-70   PAGE 2

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
70	2	104	86.66
71	1	105	87.50
72	1	106	88.33
73	1	107	89.16
74	2	109	90.83
75	1	110	91.66
76	2	112	93.33
79	1	113	94.16
80	1	114	95.00
82	1	115	95.83
89	1	116	96.66
90	1	117	97.50
91	1	118	98.33
99	1	119	99.16
104	1	120	100.00

****

TOTALS FOR FILE:

NUMBER ENTRIES EXAMINED = 500

NUMBER DATA ELEMENTS FOUND = 120

NUMBER ENTRIES WITH NO OCCURRENCE OF DATA ELEMENT = 405

NUMBER OCCURRENCES OF MULTIPLE DATA ELEMENTS = 21

MULTIPLICITY	NUMBER OCCURRENCES
2	17
3	4

****

STATISTICS CALCULATED:

MIN = 17

MEAN = 46.29

MAX = 104

VARIANCE = 354.51

STANDARD DEVIATION = 18.83

SKEW = 0.85

KURTOSIS = 0.14

*Other Added Entries-Corporate Name

**** STATISTICS FOR DATA ELEMENT SASU* **

MARC TAPE 08-13-70 PAGE

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
6	3	3	0.48
7	3	6	0.98
8	3	9	1.46
9	8	17	2.77
10	10	27	4.40
11	8	35	5.70
12	8	43	7.01
13	16	59	9.62
14	8	67	10.93
15	14	81	13.21
16	11	92	15.01
17	25	117	19.09
18	17	134	21.86
19	13	147	23.98
20	23	170	27.73
21	24	194	31.64
22	21	215	35.07
23	15	230	37.52
24	24	254	41.43
25	9	263	42.90
26	17	280	45.67
27	17	297	48.45
28	20	317	51.71
29	16	333	54.32
30	20	353	57.58
31	21	374	61.01
32	14	388	63.29
33	19	407	66.39
34	18	425	69.33
35	8	433	70.63
36	11	444	72.43
37	14	458	74.71
38	16	474	77.32
39	9	483	78.79
40	13	496	80.91
41	12	508	82.87
42	14	522	85.15
43	20	542	88.41
44	5	547	89.23
45	9	556	90.70
46	8	564	92.00
47	9	573	93.47
48	3	576	93.96
49	4	580	94.61
50	1	581	94.77

*Subject Added Entries-Subject Heading

**** STATISTICS FOR DATA ELEMENT SASU* ****

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
51	3	584	95.27
52	4	588	95.91
53	3	591	96.41
54	3	594	96.90
55	1	595	97.06
56	1	596	97.23
57	6	602	98.20
59	3	605	98.69
60	2	607	99.02
61	1	608	99.18
62	1	609	99.34
64	2	611	99.67
73	1	612	99.84
74	1	613	100.00

****

TOTALS FOR FILE:

NUMBER ENTRIES EXAMINED = 500

NUMBER DATA ELEMENTS FOUND = 613

NUMBER ENTRIES WITH NO OCCURRENCE OF DATA ELEMENT = 88

NUMBER OCCURRENCES OF MULTIPLE DATA ELEMENTS = 170

MULTIPLICITY	NUMBER OCCURRENCES
2	145
3	21
4	3
6	1

****

STATISTICS CALCULATED:

MIN = 6

MEAN = 29.02

MAX = 74

VARIANCE = 150.62

*Subject Added Entries-Subject Heading

****

STANDARD DEVIATION =   12.27

SKEW =   0.51

KURTOSIS = -0.02

*Subject Added Entries-Subject Heading

**** S T A T I S T I C S   F O R   D A T A   E L E M E N T TSRT* ****

MARC T A P E 08-13-70 PAGE 1

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
12	1	1	0.37
13	1	2	0.74
14	2	4	1.49
15	4	8	2.99
16	4	12	4.49
17	5	17	6.36
18	9	26	9.73
19	12	38	14.23
20	17	55	20.59
21	12	67	25.09
22	13	80	29.96
23	11	91	34.08
24	5	96	35.95
25	6	102	38.20
26	6	108	40.45
27	7	115	43.07
28	3	118	44.19
29	4	122	45.69
30	7	129	48.31
31	5	134	50.18
32	2	136	50.93
33	5	141	52.80
34	6	147	55.05
35	1	148	55.43
36	3	151	56.55
37	4	155	58.05
38	4	159	59.55
39	9	168	62.91
40	1	169	63.29
41	2	171	64.04
42	4	175	65.54
43	4	179	67.04
44	6	185	69.28
45	2	187	70.03
46	4	191	71.53
47	2	193	72.28
48	3	196	73.41
49	1	197	73.78
50	2	199	74.53
51	4	203	76.02
52	1	204	76.40
53	3	207	77.52
54	1	208	77.90
55	1	209	78.27
56	2	211	79.02

*Remainder of Title Statement

**** S T A T I S T I C S F O R D A T A E L E M E N T TSRT* ****

MARC T A P E 08-13-70 PAGE 2

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
57	1	212	79.40
59	1	213	79.77
60	2	215	80.52
61	2	217	81.27
62	3	220	82.39
63	1	221	82.77
65	2	223	83.52
66	3	226	84.64
67	1	227	85.02
68	1	228	85.39
69	1	229	85.77
70	1	230	86.14
71	2	232	86.89
73	2	234	87.63
75	1	235	88.01
79	1	236	88.38
80	2	238	89.13
81	1	239	89.51
83	1	240	89.88
84	2	242	90.63
86	1	243	91.01
89	1	244	91.38
93	2	246	92.13
94	2	248	92.88
101	1	249	93.26
103	1	250	93.63
104	2	252	94.38
107	2	254	95.13
116	1	255	95.50
121	1	256	95.88
127	1	257	96.25
135	1	258	96.63
137	1	259	97.00
138	1	260	97.38
140	1	261	97.75
142	1	262	98.13
144	1	263	98.50
161	1	264	98.88
163	1	265	99.25
175	1	266	99.63
214	1	267	100.00

****

TOTALS FOR FILE:

NUMBER ENTRIES EXAMINED = 500

*Remainder of Title Statement

****

NUMBER DATA ELEMENTS FOUND =   267

NUMBER ENTRIES WITH NO OCCURRENCE OF DATA ELEMENT =   234

NUMBER OCCURRENCES OF MULTIPLE DATA ELEMENTS =   1

MULTIPLICITY	NUMBER OCCURRENCES
2	1

****

STATISTICS CALCULATED:

MIN =   12

MEAN =   42.86

MAX =   214

VARIANCE = 1028.46

STANDARD DEVIATION =   32.07

SKEW =   2.17

KURTOSIS =   5.50

*Remainder of Title Statement

**** S T A T I S T I C S   F O R   D A T A   E L E M E N T   TSRT* ****

MARC T A P E 10-01-70

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
15	2	2	0.67
16	8	10	3.38
17	12	22	7.45
18	10	32	10.84
19	16	48	16.27
20	20	68	23.05
21	23	91	30.84
22	15	106	35.93
23	8	114	38.64
24	14	128	43.38
25	6	134	45.42
26	9	143	48.47
27	8	151	51.18
28	4	155	52.54
29	6	161	54.57
30	9	170	57.63
31	6	176	59.66
33	1	177	60.00
34	3		61.02
35	3		62.03
36	8	[illegible]	64.74
37	4	195	66.10
38	1	196	66.44
39	8	204	69.15
40	3	207	70.16
41	2	209	70.84
42	4	213	72.20
43	3	216	73.22
44	6	222	75.25
45	3	225	76.27
46	3	228	77.28
47	1	229	77.63
48	4	233	78.98
49	1	234	79.32
51	5	239	81.02
52	1	240	81.35
53	2	242	82.03
54	2	244	82.71
55	3	247	83.73
56	2	249	84.41
57	1	250	84.74
58	2	252	85.42
59	3	255	86.44
60	2	257	87.12
61	1	258	87.45

*Remainder of Title Statement

**** S T A T I S T I C S F O R D A T A E L E M E N T TSRT* ****

MARC T A P E 10-01-70 PAGE 2

****

LENGTH	INCIDENCE	INCIDENCE SUM	INCIDENCE PERCENTILE
62	2	260	88.13
63	2	262	88.81
64	1	263	89.15
65	1	264	89.48
66	1	265	89.83
67	1	266	90.16
70	2	268	90.84
71	1	269	91.18
72	1	270	91.52
73	2	272	92.20
74	1	273	92.54
76	1	274	92.88
78	1	275	93.22
81	1	276	93.55
82	1	277	93.89
85	1	278	94.23
90	1	279	94.57
93	3	282	95.59
97	1	283	95.93
106	1	284	96.27
110	1	285	96.61
115	2	287	97.28
125	1	288	97.63
126	1	289	97.96
131	1	290	98.30
132	1	291	98.64
137	1	292	98.98
140	1	293	99.32
148	1	294	99.66
190	1	295	100.00

****

TOTALS FOR FILE:

NUMBER ENTRIES EXAMINED = 500

NUMBER DATA ELEMENTS FOUND = 295

NUMBER ENTRIES WITH NO OCCURRENCE OF DATA ELEMENT = 205

NUMBER OCCURRENCES OF MULTIPLE DATA ELEMENTS = 0

****

*Remainder of Title Statement

****

STATISTICS CALCULATED:

MIN = 15

MEAN = 37.48

MAX = 190

VARIANCE = 690.73

STANDARD DEVIATION = 26.28

SKEW = 2.39

KURTOSIS = 6.97

*Remainder of Title Statement

APPENDIX D

SUMMARY OF A FEASIBILITY STUDY ON THE PARTICIPATION OF FOUR COLLEGES AND UNIVERSITIES IN A STANFORD UNIVERSITY LIBRARY AUTOMATION NETWORK

Summary of a Feasibility Study on the Participation of Four Colleges and Universities in a Stanford University Library Automation Network

Eleanor Montague

November 1971

SPIRES/BALLOTS Project
Stanford University
Stanford, California

ABSTRACT

Five colleges and universities met at Stanford University over a period of 13 weeks in a cooperative workshop. The purpose of the workshop was to determine the costs and benefits involved in network use of library automation and information retrieval systems (BALLOTS and SPIRES) being developed at Stanford. A feasibility analysis established the network operating environment, the impact on user-libraries of participation, the operating costs of using the system (including computing, manual, and displaced costs), and the comparative costs of performing technical processing functions manually and in the network. The results of the workshop were presented in a two-volume feasibility report to the four library administrations.

For the benefit of additional potential network libraries, the work done in the feasibility study as well as the findings are summarized in this volume, along with a detailed discussion of and how-to-do-it guide to the cost analysis methodology. A discussion of the potential impact on a library of network participation concludes the summary.

ACKNOWLEDGMENTS

The original Feasibility Study on which this document is based was the work of a team of representatives from Stanford and four other participating libraries. They were:

Karen Coates
Douglas Ferguson (Stanford)
Annette Fitzmaurice
Fumi Fowells
Lois Fullerton
Eva Kreshka
Eleanor Montague (Stanford)
Fred Siemon
Philip Warman

Peter Johnson, librarian at Stanford's Meyer Undergraduate Library, shared the work of the Stanford Cost Study.

CONTENTS

APPENDICES

LIST OF FIGURES

I INTRODUCTION

Part I covers the background, objectives, and general approach of the feasibility study and describes the organization of this summary report.

A. Background

Stanford University is developing an on-line computer system to support the acquisition and cataloging operations of its main library system. Stanford University Libraries is a large (1.8 million volumes) multilingual research collection that is growing at a rate of 80,000 volumes a year. A staff of 85 librarians and 175 library assistants make the resources of this varied collection available to the students, faculty, and staff of the university community. Stanford's library automation program is a concerted attempt to control the rising costs of acquiring and cataloging new material. This program also aims to maintain and improve service in the face of increasingly diverse demands made on the growing collection by growing numbers of academic users.

With assistance from the Office of Education, Department of Health, Education, and Welfare, the automation development effort has been underway at Stanford for four years as Project BALLOTS (Bibliographic Automation of Large Library Operations using a Time-sharing System). Working in close collaboration and sharing common software with BALLOTS is a generalized information storage and retrieval project--SPIRES (Stanford Public Information REtrieval System)--supported by the National Science Foundation. A prototype acquisition system (BALLOTS I) was implemented and operated during 1969. It was evaluated, and a production system (BALLOTS II) was defined that would be capable of reliably supporting daily library technical processing operations.

Considerable time, money, and technical skill have been invested in developing Stanford's on-line technical processing system. The cost and effort involved in on-line system development suggest that the resulting system should be used as widely as possible. BALLOTS will be implemented as a series of modules (sets of services). The first module, BALLOTS-MARC, is the focus of this report. The BALLOTS-MARC module was given highest priority for early implementation because it would be broadly useful both to Stanford and to other academic libraries.

To promote the broadest possible use of BALLOTS, Stanford invited library directors from 12 nearby colleges and universities to meetings in July 1970 and February 1971. The participants in these meetings explored the possible usefulness of BALLOTS to other libraries. It was agreed that a preliminary feasibility study was a prerequisite to actual participation in BALLOTS. At the February meeting, four schools agreed to commit the necessary personnel and time to this study. Two senior librarians with technical processing experience were selected by each school. They met together with members of the BALLOTS staff at Stanford one day a week for 13 weeks from April to July. The equivalent of a second day each week was spent by team members at their own schools gathering data for a cost analysis. At the conclusion of the study, a two-volume, 250-page feasibility report <1>, containing the findings for all four libraries, was prepared and distributed to the directors of the four libraries.

B. Feasibility Study Objectives

The three-month feasibility study had several objectives.

1. To learn how BALLOTS-MARC will support library technical processing.

2. To develop a cost measurement methodology that would be applicable within the time limits of the study. This methodology had to be uniformly applicable without placing a heavy data-gathering burden on the staff of each library; it therefore permitted estimates where firm data was not readily available. It had to permit comparison of the costs of existing manual procedures and the costs involved in the use of BALLOTS-MARC <2>.

3. To establish the costs to each participating library of processing their library searches and producing printed outputs through BALLOTS-MARC, and to establish their computing overhead charges.

4. To prepare a preliminary design of work flow for each library showing how BALLOTS-MARC would be used.

5. To discover ways in which BALLOTS-MARC could be made of greater use to libraries outside of Stanford. Suggestions would be passed on to the BALLOTS management.

6. To prepare a report that would provide cost and benefit information for the director of each library. This report was to enable each library to determine the main implications and results of its participation in a BALLOTS regional network.

C. General Study Approach

The study team met as a group at Stanford for six hours every Friday for 13 weeks. At the first meeting, team members described their backgrounds and interests, the features of BALLOTS were reviewed, and each library was generally described by someone from that library. During subsequent sessions a cost measurement methodology was developed and applied. At various times, BALLOTS staff members spoke to the group on background topics, such as the characteristics of video display terminals in an on-line library system.

1. Manual Costs

The major tasks performed in the course of the study included the following. Library technical processing was conceptualized as a series of functions (ordering, card production, etc.), each consisting of several activities (searching catalogs, typing purchase orders, typing card headings, card filing, etc.) to which personnel times could be assigned and for which, in consequence, personnel costs could be calculated. Equipment and supply costs were allocated over the entire function. For each function a flow chart was prepared and a measurable unit of production was determined (number of purchase orders, number of catalog cards typed, etc.). A Function Worksheet and a Function Summary Form were designed and then tested and revised (see Appendix A). The worksheet permitted the personnel times and costs to be collected and calculated for every activity in a function, and the summary form displayed a summary of all the major costs of a function.

2. Manual-Automated Costs

After manual costs were calculated for each function, the functions were flow-charted for each library showing how they would be carried out using BALLOTS-MARC for searching and output production. Statistics were collected in order to establish the volume of each library's search and output transactions. BALLOTS-MARC processing costs (on-line and batch charges) were calculated from these transaction estimates. Costs for dedicated equipment (e.g. a terminal) used by a single library were assigned to that library. Overhead costs were assigned to each library on a percentage basis. The costs of each function affected by BALLOTS-MARC were then recalculated, with on-line and batch computer processing costs replacing some manual processing costs. Personnel times for the new manual-automated activities were standardized for all libraries. The function costs based on manual processing could then be compared with those based on manual plus automated processing. Computing overhead costs were not allocated to functions, but were treated as a separate cost to be added to MARC function costs.

D. Purpose of the Summary Report

The two-volume feasibility report recorded the tools and techniques for the feasibility analysis developed during the cooperative workshop, and it reported on the findings for the four participating libraries. After the report had been issued, the same cost analysis methodology was used, with refinements and extensions, in the Stanford University Libraries. The Stanford Cost Study was issued as a separate library report.

This summary is directed primarily to any other library interested, even remotely, in joining a regional library automation network and in performing its own feasibility study. The summary is in four parts. Part I highlights the original workshop objectives and approach and describes the organization of the summary report. Part II describes in detail the BALLOTS-MARC module (the first set of services provided by the system) as it will operate in 1972. Part III presents a detailed how-to-do-it guide to the BALLOTS cost analysis methodology. Using this part, any library could set its own personnel to conduct a complete cost analysis of current technical processing activities and of the same operations under BALLOTS-MARC. Finally, Part IV summarizes the conclusions and cost findings of the participating libraries and Stanford.

This report concentrates on the BALLOTS-MARC module, the first of the 11 modules to be implemented.

II BALLOTS-MARC MODULE

This part describes the BALLOTS-MARC module as it will be implemented at Stanford in March 1972. The network implementation of the MARC module will begin September 1972. An overview is presented first, followed by a discussion of the development schedule, the system in more detail, the step-by-step processing of a book, the costs to user-libraries, and planned extensions to BALLOTS-MARC.

A. Summary of the Module

BALLOTS-MARC is the first part to be implemented of an automated system (BALLOTS II) to serve the technical processing operations of Stanford University Libraries. BALLOTS II is an on-line production computer system using video display (also called CRT--cathode ray tube) terminals in the library. Since the system is ON-LINE, a librarian will be able to search a machine-readable file (such as the MARC file), add information to a record, or check the status of a record at any time during the working day. In a BATCH system, by contrast, all searches or record changes are grouped together and processed by the computer at one time (often at night), along with other jobs. The user then receives a printed output the next morning describing the results of the batch processing. In contrast, an on-line system is available to the library staff moment by moment during their work, so that it becomes as common and frequently used as a typewriter or a copy of Cumulative Book Index. BALLOTS II is a PRODUCTION system because it is designed to handle the sizeable volume of file- and record-keeping activity in technical processing with the assurance of reliable service and protection against the destruction of machine-readable records. The use of video display terminals gives the librarian a fast, silent source of information to assist him in his work, as opposed to the slower, noisy, character-by-character printing of the typical typewriter terminal.

BALLOTS-MARC is designed to provide on-line access to recent MARC records within a few days after they have been received from the Library of Congress (LC). MARC records (the cataloging information supplied by LC) will be used in book ordering and cataloging by Stanford and other user-libraries. The bibliographic information that is already in machine-readable form will not have to be typed manually in preparing purchase order forms, forms for various in process files (e.g. a vendor file), and catalog cards.

The main file in BALLOTS-MARC will consist of about 50,000 to 100,000 MARC records produced during the most recent 6- to 12-month period. MARC records now cover all English-language books cataloged by the Library of Congress each week. By the end of 1972, this coverage will have expanded to French, Italian, Spanish, Portuguese, and Romanian, and by early 1973 to German. A librarian will be able to search the BALLOTS-MARC file by LC card number, personal, corporate, or conference name, or title word. Each of these search keys is an index to the file. With the bibliographic information in a MARC record, a purchase order can be produced by adding ordering information; catalog cards can be produced by adding locally required data, such as holdings information or added subject headings. When a purchase order is produced, appropriate technical processing control forms can also be printed. When catalog cards are produced, spine labels can be printed from the same data.

It is important to realize that BALLOTS-MARC operates in parallel with manual ordering and cataloging operations, which handle any material falling outside the scope of BALLOTS-MARC. BALLOTS-MARC services only titles covered by MARC cataloging (e.g. English-language monographs). BALLOTS-MARC does not handle the payment process, claiming, or the cancellation of orders. Section II.E, "Extensions to BALLOTS-MARC," describes the broader range of material that future BALLOTS modules will handle.

B. Development Schedule

BALLOTS-MARC is scheduled to be implemented in the main library of Stanford in March 1972. The initial network is scheduled to be implemented in September 1972. At the present time, library requirements for inputs, outputs, and system features have been defined by system analysts and librarians. Programs are being written and debugged from July to December 1971. The first video display terminal was delivered in September, and it will be ready for use in testing programs with real library data by November. BALLOTS is expected to be one of the first users of the new Sanders 800 series upper/lower-case terminal (see Figure 1).

C. Module Description

The following is a more detailed description of BALLOTS-MARC. BALLOTS-MARC will use the Stanford Computation Center's IBM 360 Model 67 computer, and the MARC file will be stored on a disk storage device. Upper/lower-case video display terminals (the Sanders Associates 800 series) will be in the library. A smaller computer, a PDP-11, will handle communication between the library terminals and the 360/67.

1. Processing Functions

There are four major processing functions in BALLOTS-MARC. A function maybe thought of as work accomplished by the computer or by the user and the computer together. One function creates

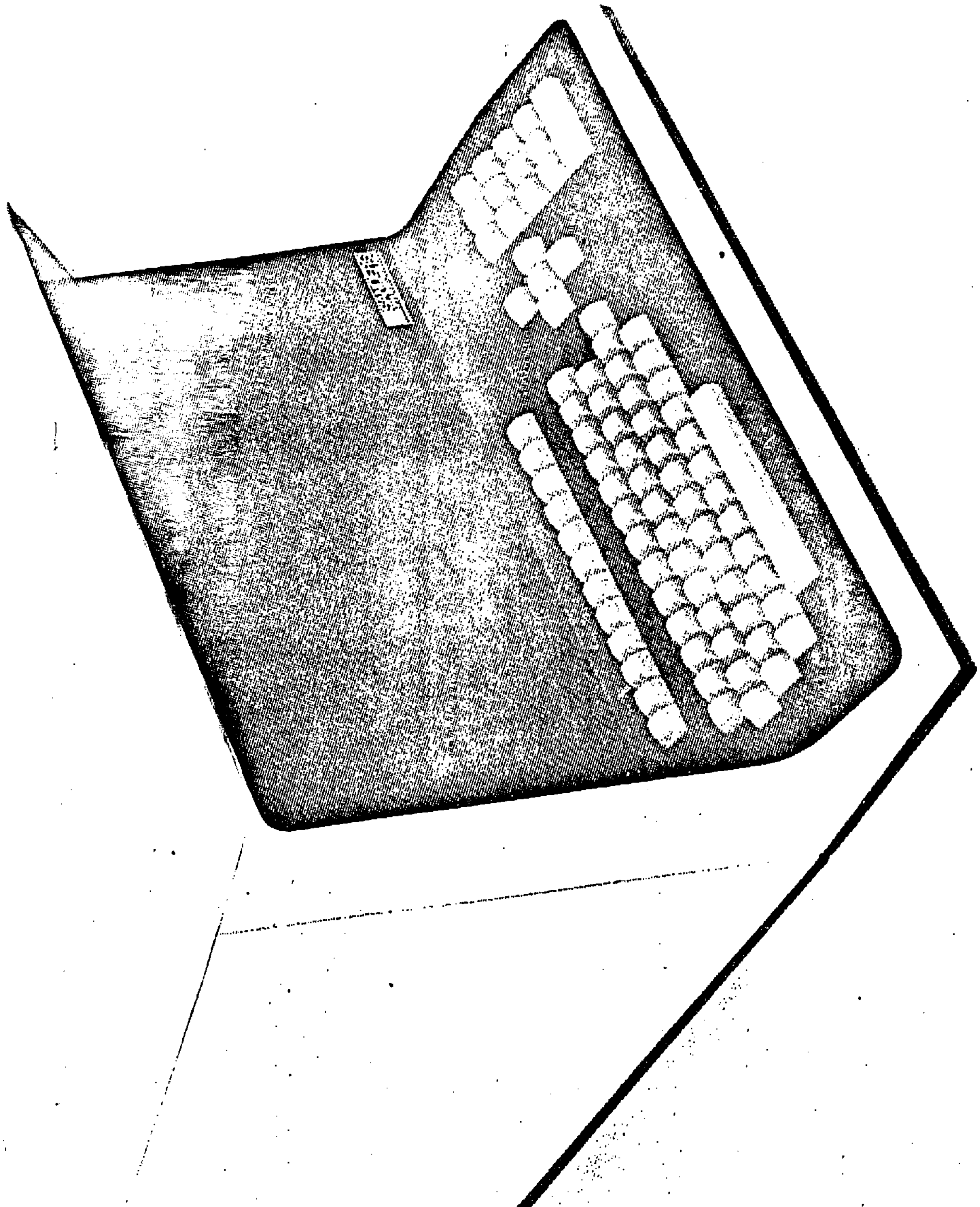


Figure 1. Sanders 800 Series Visual Display Terminal

and maintains the MARC file, a second produces purchase orders and related book processing forms, a third produces catalog cards and spine labels, and a fourth permits an automatic periodic file search for a specified record. The BALLOTS-MARC functions are called file maintenance, ordering, cataloging, and standing search.

MARC file maintenance is batch processing carried on during hours of low system activity (usually at night) and does not directly involve users. Computation Center personnel handle all operations according to instructions prepared by the BALLOTS staff. Ordering and cataloging combine on-line processing of searches and on-line entry to a record of new or modified information with batch processing of printed outputs. Users (librarians), of course, are directly involved in the on-line aspects of these functions. Standing search is initiated on-line by a user and done in batch mode each week in conjunction with file maintenance processing. In effect, it permits the user to request that additions to the file be searched weekly until either a given record is found or the request is cancelled.

2. Function Tasks

The initial MARC file is created by converting a large group of MARC tapes from the MARC II format to the BALLOTS II format. This conversion is necessary each time a tape is added to the MARC file, because MARC records are not directly searchable and displayable by BALLOTS-MARC in the MARC II format. MARC tapes, consisting of 1,200 to 1,800 records, are received each week and are added to the MARC file. At the same time that this weekly addition of records is made to the file, another large group of records is deleted from the file. The deleted records are those records that are older than a predetermined date. If a record is involved in some technical processing activity, its deletion date is automatically extended. As we add records to the MARC file each week, then, we drop an approximately equivalent number from the file in order to maintain a constant size.

The ordering function handles on-line searches initiated by a user in the process of ordering a book. It also handles the collection of ordering information and the subsequent printing of purchase orders and related forms. The cataloging function handles on-line searches for material received that is within the scope of MARC. The record is updated to make any necessary changes to the bibliographic information, and holding information is added. A request is entered for catalog cards and spine labels, and these are subsequently printed in an overnight batch run.

The standing search function is used for a title not found in the file but for which a MARC record is expected. It may be initiated during ordering or after the book is received. When a title cannot be found in the MARC file, a request for a standing search is entered. This is processed weekly during file maintenance until the record is found or the request is deleted.

3. Ordering and Cataloging a Title

The following is a step-by-step description of the way a library would acquire a book and make it available for circulation using BALLOTS-MARC. The user in the following description is a librarian or a library assistant.

Requests for books are checked to be sure that the books are not on order or already in the collection. The requests that are to be filled by purchasing books are separated into those for which a MARC record is likely to exist and those for which a MARC record is not likely. The user then sits down at a terminal with a number of requests of the first type and searches the MARC file. He begins by going through a "sign on" procedure in which he identifies himself, the library, the function that he is now performing (ordering), and the file to be searched (MARC). This identifying information need not be entered again during the on-line session, unless there is some change in status, such as a change in user or a desire to perform another function such as cataloging. The search screen format is immediately presented, and the user types in a search request by the author, title, LC card number, or any combination of these. This and each successive screen are transmitted to the computer for processing by pushing a "send" button on the keyboard. The search request may look as follows:

find author Reich and title Greening of America

A response is displayed on the screen almost instantaneously (two to four seconds). The response to a search request contains the original request and a simple statement of the number of records found. Usually this will be one, since it is expected that most technical processing searches will be based on fairly accurate bibliographic information. If more than one record is found, the user may engage in an interactive search by supplying more title words, another author, etc.

If a unique record for the request has been located, the MARC record (i.e., the full bibliographic information) is automatically displayed. This gives the user a chance to verify that the record is for the book to be ordered and to spot-check whether or not certain critical information needs to be modified. It is expected that changes will be rare but could be needed--for example, if the American edition of a work is wanted and the MARC record is for the British edition. If the record needs to be modified, the user requests that the record be displayed on the bibliographic input screen, which permits information to be modified. The data is changed by typing over it, displacing the original data, and the modified record is transmitted for processing. The user then calls for an empty ordering screen on which he fills in all the information needed to order a title. This information is transmitted along with a purchase order print request. When the purchase orders have been printed, overnight, they are picked up from the Computation Center and delivered to the library, where the appropriate number of copies as stipulated by the user-library is sent to the vendor and file copies are retained. Stanford uses an IBM-card-sized purchase order form, as illustrated in Figure 2.

As ordered books are received, it will be possible to identify those ordered using BALLOTS-MARC either by the purchase order copy accompanying the book or by the library's in-process record of the order. Suppose that the book ordered in the above paragraph has arrived and been identified as a MARC book. Since the LC card number is available, the search request on the search screen will be easily formulated as follows:

find crd 73-103412

If the user wishes to change the bibliographic data, he asks for an input screen. When the bibliographic data is acceptable for printing on a catalog card, the user calls the holdings screen. The LC call number is modified if necessary. Location and physical part information (vol. 1, copy 2, etc.) are added, and then the command to print catalog cards is entered. Spine labels are printed on a typewriter terminal. (For Stanford the typewriter terminal will be in the library; for other user-libraries spine labels may be printed centrally at the user-library terminal or at a terminal in the computation center.) Catalog cards will be printed in a predetermined order or in several orders to aid in filing. Figure 3 presents Stanford's catalog card design.

D. Cost to User-Libraries

The method used to calculate costs is given in Part III. Here we will review general cost considerations that apply to all user-libraries.

As detailed in Part III, there are four classes of BALLOTS-MARC costs: (1) on-line transaction charges, (2) batch transaction charges, (3) dedicated equipment charges, and (4) computing overhead charges. An on-line transaction charge is made for each file search or change in a record requested by a user at a video terminal. A batch transaction charge is made for each printed output resulting from an on-line transaction. Equipment charges cover terminal rental, data set rental, and telephone line charges. Overhead charges cover shared file storage, connection time, and major file updating. The major computing overhead charge is maintaining the MARC file; that is, adding a new tape each week and deleting old records.

STANFORD UNIVERSITY LIBRARIES

P U R C H A S E O R D E R

[redacted]

DATE OF ORDER	ORDER NO.
4-29-69	2581

NO. COPIES: 1

Tarski, Alfred, 1902-.
Undecidable theories. In collaboration with Andrzej Mostowski and Raphael M. Robinson. 2d printing ed. Amsterdam, North-Holland, 1968.

(Studies in logic and the foundations of mathematics.)

SHIP AND BILL IN DUPLICATE TO:
Order Department
Stanford University Libraries
Stanford, CA 94305

TOTAL EST. PRICE ▶ Fl 15.   VEND. NO. 30

XXXXXXXXXXXXXXXXXXXX

DEALER LEAVE BLANK ▶ INV. DATE | L.P. | N.P. | S. TX. | S.P.

DEALER: SEE OTHER SIDE

SUL-200 (1-69)

VARCO BUSINESS FORMS · OAKLAND, CALIF. 271347

Figure 2. Stanford University Libraries Computer-printed Purchase Order Form

QA76.5.M184
McCracken, Daniel D
Digital computer programming.

<u>Additional locations for the item described on next card(s) are listed below:</u>

PHYSICS
COMPUTER SCI.
ENGINEERING
HANSEN LABS.

70-12345

QA76.5.M184 STACK
McCracken, Daniel D
Digital computer programming. New York, J. Wiley, 1957.
253 p. illus. 24cm. (General Electric series)

1. Electronic digital computers - Programming.
I. Title.

70-12345

QA176.5.M184 STACK
ELECTRONIC DIGITAL COMPUTERS - PROGRAMMING.

McCracken, Daniel D
Digital computer programming. New York, J. Wiley, 1957.
253 p. illus. 24cm. (General Electric series)

70-12345

QA76.5.M184
COMP: c.3,8
ENG: c.5-7
HANS: c.4
PHYS: c.1
STK: c.2,9

McCracken, Daniel D
Digital computer programming. New York J. Wiley, 1957.
253 p. illus. 24cm. (General Electric series)

OSL 70-12345

Figure 3. Stanford University Libraries Computer-Printed Catalog Cards

There are several additional costs that libraries must consider. Space for the CRT terminal and a working table will have to be provided. There is a charge for installing the telephone equipment and the terminal. Staff personnel will have to be trained to use the terminal. Arrangements will have to be made for delivering the printed outputs; this may mean added postage.

E. Extensions to BALLOTS-MARC

The following four modules are major extensions to BALLOTS II during 1972 and 1973. The schedule estimates for these were made in June 1971. The first module, BALLOTS-MARC, will be available at Stanford in March 1972 and it will be available to the network in September 1972. All the services of previous modules are still available as new modules are implemented.

MODULE: MARC-IPF.

FILES: MARC as in first module. Adds an In Process File (IPF).

FEATURES: In addition to the on-line MARC file from the first module, the user-library will have an on-line In Process File for items in process for which a MARC record is found. The full record for a title in process can be transferred from the MARC file to the IPF and retained throughout technical processing. A record's status can be determined on-line and the file searched or updated at any time throughout the day. Claim and cancellation notices can be computer-produced on command.

SCHEDULE: Stanford, July 1972. Network, March 1973.

MODULE: Purchase Order/Original Cataloging.

FILES: MARC and IPF as above with enlarged IPF scope.

FEATURES: The user-library will have an In Process File (IPF) for all titles acquired by purchase order. Thus, even if the title is not in MARC or even if it is necessary to perform original cataloging, the user-library will be able to create and store the record (both bibliographic and ordering data). The librarian can have computer-produced NPAC (National Program for Acquisitions and Cataloging) notices and Title II slips.

SCHEDULE: Stanford, November 1972. Network, June 1973.

MODULE: Non-Purchase Order (NPO).

FILES: MARC and IPF as above. No new files.

FEATURES: The user-library will be able to create an IPF record for a title acquired by blanket order, as a gift, or by exchange for which no MARC record was found. NPO material with a MARC record is included in the first module and following modules. A list of books received without invoices will be produced regularly.

SCHEDULE: Stanford, December 1972. Network, August 1973.

MODULE: Catalog Data File (CDF).

FILES: MARC, IPF as above, and adds CDF.

FEATURES: The user-library will be able to store on-line a record of each title acquired by the library after implementing the BALLOTS-MARC module. A copy of all bibliographic and holdings information will be retained on magnetic tape for all acquisitions processed since implementation of the first module. This tape file will be used to create the initial CDF. From that point on, as a title completes technical processing, its record can be transferred from the IPF to the CDF, where it is accessible to the library staff, to library patrons, and to other members of the network.

SCHEDULE: Stanford, January 1973. Network, November 1973.

Further technical processing modules, such as Automatic Claiming and Cancellation and Standing Orders will be implemented at Stanford in 1973 and for the network in 1974. Circulation modules involving an additional small computer (installed in the library) and on-line files will be implemented at Stanford in 1973. Circulation services for the network will be investigated, and a preliminary design, hardware configuration, and cost estimates will be presented to any network library interested in participating in this study.

III LIBRARY COST ANALYSIS

A. Cost Study Methodology

The cost methodology was tailored specifically to measure the effects on technical processing costs of using the first module of Stanford's library automation (BALLOTS) and information retrieval (SPIRES) system <3>. As a result, the methodology gives a library director all the quantitative data necessary in order to evaluate the use of BALLOTS-MARC: the expected dollar decrease in personnel costs accountable to automation and work simplification; the costs of on-line and batch transactions (high and low); the CRT equipment and associated equipment rental charges; and the computer overhead charges. Of these costs, only the proportion of the computer overhead charged to each library might vary owing to external circumstances--namely, the number of libraries in the network. The methodology can be used by any library that is interested in joining the Stanford network and desires cost data for management decision making.

The methodology is straightforward and pragmatic. It relies on library personnel <4> to do the analysis, using existing library statistics as far as possible for all cost calculations. There is no stopwatch or pedometer measurement; and work sampling and observation studies are not encouraged in normal situations.

In general the approach, which has been described in I.C, is (1) to divide technical processing into logical discrete functions (see III.A.1); (2) to identify the component activities within each function for which costs can be calculated; (3) to calculate manual total function costs; (4) to collect quantitative data on the volume of search and output transactions--such as purchase orders--in each function, to be used in calculating BALLOTS-MARC costs; (5) to recalculate the costs for the functions that would be automated under BALLOTS-MARC, as they would then be performed; (6) to compare the manual costs of each function with the manual-automated costs under BALLOTS-MARC; and, finally, (7) to assess the benefits and drawbacks, both economic and non-economic, of automation and of the network approach.

1. The Concept of Functions and Activities

Library technical processing can be viewed as a series of FUNCTIONS, or of major segments of work to be performed. Breaking up a large section of library operations into functions makes understanding and analyzing the operations easier. This does not prevent also taking a synthetic view of the operations, in which factors relating to the process as a whole (such as the budget of an entire department) are considered.

What is a function? It is not identical with a job title or position, nor with a person or an organizational unit. In general, a function is a series of actions that are performed together in a logical sequence, and that produce an identifiable and countable result--a sorted set of catalog cards, for example. The identifiable and countable result is often called an "output," and what the series of actions starts with is often called an "input." The individual output of a function is the item to which a unit cost of production can be assigned. Thus it is possible to calculate the total function cost of ordering a book, for example, and to assign an approximate unit cost for producing a purchase order, the principal output of the Ordering Function.

We call the component actions of a function ACTIVITIES. Activities are not always easily distinguished, and in different libraries, functions that produce the same output may consist of different sets of activities. For example, some libraries type a book pocket at the end of technical processing, as part of the End Processing Function; in other libraries this activity may occur in another function.

The personnel time and therefore the main cost factor of an activity in a function can be calculated. By breaking up a function into component activies, not only can we be more certain that we have accounted for all that goes into producing a function output, but, for a small, well-defined unit of activity, we can also measure costs more accurately. Figure 4 is a sample flowchart showing the activities and outputs in the Manual Ordering Function.

2. Personnel Cost Calculations and Assumptions

An activity is the smallest segment of work for which a cost can be calculated. An activity's cost equals the time taken to perform the activity times the real cost of the people performing it. Other costs, such as supply costs, machine costs, supervisory costs, etc., cannot be conveniently allocated to a single activity and are thus allocated to the function as a whole (see III.A.4).

Because of significantly different rates of pay, personnel costs for each activity are calculated for three categories of employees: professionals, assistants, and students. The key constant factor in each library's activity cost calculations is the adjusted hourly pay rate for each category of personnel. This represents the real rate for productive working hours and is usually higher than the standard hourly rate (determined by dividing gross pay by the regular forty-hour or 38-hour work week). In this way, the cost to the library of nonproductive hours is allocated across the cost of performing productive activities, resulting in more accurate costs <5>. The adjusted hourly rate is calculated as follows.

(a) ANNUAL PRODUCTIVE WORKING HOURS--APH. Multiply the number of hours in the standard work week times 52 weeks and subtract vacation hours, paid library holidays, an average number of annual sick leave hours, and an average number of hours per year for coffee breaks. Do this for professionals and assistants. There is no APH for students.

For example, if the standard work week of a professional is 38 hours, if the standard vacation is two weeks (76 hours) per year and the average sick leave is five days (38 hours), if there are five paid holidays (38 hours), and if the professional takes two 15-minute (.5 hour) coffee breaks each working day, the the APH for professionals at this library is

(38 x 52) - (76 + 38 + 38) = 1,824 hours (or 240 days)

- 120 hours (breaks) = 1,704 productive hours per year

If reliable statistics are available to compute average time spent on staff association meetings, personal business, lateness, etc., use this information in calculating APH. Do not adjust the APH for a fatigue factor.

(b) AVERAGE ANNUAL SALARY--AAS. For each category of personnel, calculate an average annual salary and add the standard percentage for benefits (retirement, insurance, etc.). For example, if the actually paid salary ranges for professionals in technical processing positions are

Librarian I	$6,000	-	$8,000
Librarian II	$8,000	-	$10,000
Librarian III	$11,000	-	$12,000

then the average of $7,000, $9,000, and $11,500 is $9,166 or, rounded, $9,200. If the standard benefit rate is 15 percent, $1,380 is added to this average to produce a professional AAS of $10,580 <6>.

Average the salary ranges only for the positions that are actually represented in technical processing. For example, do not include an average salary for a circulation assistant in the calculations for technical processing assistants.

If the resulting AAS for a category of personnel will be more accurate, perform AAS calculations for acquisition and cataloging separately to obtain an AAS for each. If only one personnel level in a department is affected by the proposed automation, include only the AAS for that level in the averaging.

Do not include top administrators' salaries (professional or assistant) in the calculation of AAS.

(c) ADJUSTED HOURLY RATE--AHR. The adjusted hourly rate is determined by dividing the average annual salary by the annual productive hours.

AHR = AAS/APH

Calculate student AHR by averaging student hourly rates in technical processing. Add in an amount for benefits if benefits are paid for students.

3. Supplies and Equipment Costs

Only the supplies and equipment costs that would be affected by BALLOTS-MARC are included; any costs that would remain constant are excluded. Following are sample guidelines for distinguishing costs.

(a) Exclude administrative and physical plant overhead costs.

(b) Make no adjustment for the purchase price of machinery, the cost of maintenance contracts, or average repair costs.

(c) Exclude the cost of general office supplies such as stationery, typwriter ribbons, carbon paper, pencils, printed forms, etc.

(d) Include the purchase price of LC cards and proofslips. Do not include the purchase price of reference works or printed catalogs.

(e) Include supplies and equipment costs and charges for machinery involved in catalog card duplication or production.

(f) Include the per-shot cost of Polaroid shots or of pictures from a bibliographer's camera.

4. Function Worksheet and Function Summary Form

The manual cost analysis involves two standard forms. The Function Worksheet is used to collect time analysis statistics and to calculate costs for each category of personnel for every activity in a function. A Function Summary Form is used to display in summarized form all the function costs, personnel times, supplies and equipment costs, and unit costs. Appendix A contains the forms and instructions for their use.

B. Manual System Cost Calculations

Analysis begins with measuring manual processing costs and verifying the total cost against departmental budgets to ensure that the costs are reasonably accurate.

1. Defining and Charting Functions and Activities

The first task is to divide technical processing into functions. Below is a list of the 13 major functions used in the feasibility study, with a brief explanation of each. The term "book" is used in its generic sense as referring to a variety of material acquired by libraries.

(a) Ordering. The process of receiving a request to purchase a book, searching for it, and performing all the other activities that culminate in mailing a purchase order. Includes only books in print.

(b) Purchase Order Material Receipt. The process of receiving books for which a purchase order has been issued and preparing them for transmittal to cataloging.

(c) Non Purchase Order Material Receipt. The process of receiving and deciding to keep a book obtained through exchange, on approval, or as a gift. A purchase order has usually not been issued for acquisitions of this type.

(d) Distribution. In either the Acquisition or Catalog Department, the process of preparing books for distribution to any other organizational unit.

(e) Obtaining/Maintaining LC Data. The process of ordering Library of Congress (LC) cards, or of receiving LC cards and proofslips and filing them in a public catalog or other files.

(f) LC Cataloging. The processing of books with LC copy.

(g) Card Production. The process of receiving a final card copy from a cataloger or library assistant and turning it into a sorted set ready for filing into the public catalog or other files.

(h) End Processing. The process of readying a fully cataloged book for transmittal to the Circulation Department.

(i) Claiming. The process of recognizing the need to claim purchase order materials and preparing the claim.

(j) Cancelling. The process of cancelling a purchase order.

(k) Added Copies/Added Volumes. The process of adding copies and volumes to already held titles.

(l) Original Cataloging. The process of cataloging books for which no LC copy is available.

(m) File Maintenance. The process of making the card files consistent with all transfers, cancellations and corrections.

The scope of each function may vary from library to library. Not all these functions may need to be considered; and, if necessary, new functions can be added. Technical processing should be divided into functions and the functions into activities in such a way that (1) the benefit derived from currently available library statistics is maximized and (2) the functions and activities that will be affected by BALLOTS-MARC are separately defined and measured. Therefore, before actually starting on manual costs, one should read carefully Part III.C., BALLOTS-MARC System Cost Calculations, and define manual functions and activities accordingly.

After functions have been identified, it is suggested that a simple flow chart be prepared for each. These provide graphic representations for discussion and review with library personnel and help to identify all activities in their proper sequence. The charts should be simple, informal diagrams with each rectangular box representing an activity. Name all the outputs, but show only major decision points. See Figure 4 in III.A.I for an example. Prepare such flow charts for all manual functions after having reviewed the BALLOTS-MARC system and before starting cost calculations <7>. On the basis of the chart, decide on the unit of output to be measured for each function.

Five of the functions listed above are not supported by the BALLOTS-MARC module: these are Claiming, Cancelling, Added Copies/Added Volumes, Original Cataloging, and File Maintenance. The calculation of manual costs for these functions can be deferred until future modules are ready or it can be done along with the rest of the manual calculations. Determining all the manual technical processing costs at once has the advantage of supplying valuable statistics for management use, whether or not the decision is made to join the network.

2. Performing the Manual Cost Study

Once the functions and activities are defined and charted, the next step (and the hardest) is to determine the amount of pure productive time spent by professionals, assistants and students in accomplishing each activity. This time, then, when multiplied by the adjusted hourly rate for the category of personnel performing that activity over a year, yields the true cost to the library of having that activity performed. The sum of all activity costs plus supplies and equipment costs equal the total cost for that function.

Calculate the personnel costs for each activity using the Function Worksheet (refer to Appendix A) as follows:

Column C: The total hours per week that the activity is performed, even if by more than one person. MULTIPLIED BY

Column D: The number of weeks in the year that the activity is performed. RESULTS IN

Column E: The hours per year that the activity is performed. MULTIPLIED BY

Column F: The adjusted hourly rate for the personnel category for the activity. RESULTS IN

Column G: The adjusted yearly personnel cost. The sum of column for each activity (i.e., lines 1,2, and 3, the costs for each personnel category activity) is the total activity cost (line 4).

Include supervisory time in column C only if the supervisor contributed time to the performance of the activity. Under normal library conditions, all the technical processing functions and activities should be "ongoing"--that is, performed continually throughout the year, regardless of personnel absences, etc. Thus, column D usually contains the factor 52. For special projects of limited duration that are going to be included in the cost analysis, some smaller number of weeks can be used.

Modify the Function Worksheet as necessary to accommodate the form of available statistics. For example, if time analysis statistics are already available and are summarized yearly, record those yearly figures in column E (after verifying their accuracy or likelihood) and neglect the total hours per week and weeks per year in columns C and D.

Function costs are summarized on the Function Summary Form (see Appendix A). These include total personnel costs, total costs for equipment, supplies, etc., personnel time in hours per week, and in hours per year. The total function cost is divided by the annual units produced (IV), such as volumes ordered or card sets sorted; the result is a unit cost for a function output (e.g., cost per card set, per purchase order, or per Library of Congress title cataloged).

As stated above, it is assumed that this cost study will be done by library personnel in short order, to collect the data needed to evaluate the effects on personnel time and costs of using BALLOTS products and services. In most libraries, time constraints require that time data be derived from currently available statistics, from interviews with staff members, or from both. If a library wants to use the occasion of this cost study to initiate a time analysis study on a sample or long-range basis, there are books and articles that can help in designing the forms needed, etc. But if such a data-gathering task is initiated, it should be designed so that the findings answer the requirements of the cost analysis study.

3. Verification of Cost Results

As a check on the overall accuracy of the manual costs determined, compare the total of all the function costs in a department with the budget of that department for personnel (including staff benefits), equipment, and supplies. Since by definition the functions considered are not all-inclusive, it is necessary at this point to account for the areas of activity outside of the functions. Examples include departmental supervision, recataloging, reclassification, and out-of-print procurement.

Identify and briefly describe each area of technical processing unaccounted for, and roughly estimate the personnel costs for each. Compare the sum of these total costs plus the sum of all total function costs (see the Function Summary Form) with the departmental budget. A plus or minus 10 percent difference is generally acceptable.

C. BALLOTS-MARC System Cost Calculations

In the second phase of the cost analysis, computer costs are determined and the manual costs for the functions affected by BALLOTS are recalculated. This recalculation measures the cost effect on the activities that remain manual but are affected by automation, and adds costs for the new manual-automated activities.

1. BALLOTS-MARC Functions and Activities

Part II (especially II.C.3) of this summary describes the general capabilities and outputs of the BALLOTS-MARC module without referring to functions or to library departments. In order to decide exactly how these capabilities and outputs will be implemented in a given library organization, it is necessary to re-chart each function, showing how it would be performed under manual-automated conditions using BALLOTS-MARC. The new flow charts must show all the continued manual functions as well as the new manual-automated functions. Figure 5 is a sample diagram of the Manual-Automated Ordering Function as it might look in a library. As in the manual diagrams, all activities and outputs are named and major decisions shown. The manual-automated activities and computer-printed outputs are crosshatched for easy identification.

2. Library Transaction Volume Estimates

Before actually starting on the cost analysis, it is helpful to estimate the volume of acquisition and cataloging activity and outputs that will be handled through the BALLOTS-MARC module. To simplify and standardize this procedure, a statistics questionnaire is used; it asks for specific figures on such things as the number of titles for which MARC is expected during ordering and cataloging, the number of catalog cards, etc. A schematic diagram of the module, keyed to the questionnaire, is included with the questionnaire in Appendix B. For the purposes of the questionnaire, assume an on-line MARC file containing records covering the latest year's worth of Library of Congress cataloging for English-language monographs. It is not easy to complete the questionnaire, since library statistics simply do not exist that will specifically answer the questions. However, educated guesses based on what statistics are available will be sufficiently accurate for this cost analysis.

3. Computing Costs

There are four kinds of costs involved in the use of a computing system. Users need to understand what these costs are and how they are calculated. They are essentially the costs for which a user will receive a bill each month from either equipment manufacturers or BALLOTS:

(a) On-Line Transaction Costs--for activities involving use of a video terminal, such as searching for or modifying a record.

(b) Batch Transaction Costs--for printing purchase orders, catalog cards, and so on.

(c) Dedicated Equipment Costs--for terminal rental and telephone line charges.

(d) Computing Overhead Costs--for shared use of file storage, updating, and connection charges.

Transactions are the actions performed by a user at a terminal to send information to the computer and cause some processing of this information. BALLOTS-MARC unit transaction costs and overhead charges are given in Appendix C.

(a) On-line Transactions. In BALLOTS-MARC, the two typical on-line transactions are (1) searching to locate a record and (2) modifying a record by adding or changing data to produce a printed output. Since each transaction involves some computer processing, the unit cost of a transaction is determined by the amount of time that the Central Processing Unit (CPU) takes to handle it. CPU cost may be several hundred dollars an hour; but because of the speed of computers and the time-shared software of the 360/67, a transaction can be handled in a few thousandths of a second. Thus, the unit cost of a transaction ranges from 2 cents to 33 cents. To provide a margin of safety for user-libraries, high and low (or worst and best case) figures are used to calculate transaction costs.

To reflect variation further, unit costs for simple, medium, and complex searches are calculated. A simple search asks for one item by a discrete identifier using one index:

find LC card number 73-103412

A medium search uses more than one index:

find author Smith and title History

A complex search uses a search query with several logical operators, and a dialogue may be involved:

find author Smith or Jones and title History
6 BOOKS FOUND
title french
1 BOOK FOUND

These are only samples of the ways in which simple, medium, and complex searches could be stated.

(b) Batch Transactions. Batch transactions produce printed library outputs. The outputs to be printed through BALLOTS-MARC are purchase orders, cards, processing slips, and spine labels. Batch transactions also handle the standing search outputs for matched or purged requests. As with on-line transactions, the unit cost of each batch transaction is based on the fraction of a second of CPU time taken. In addition to the processing costs for batch transactions, there is a print charge for each item printed.

(c) Equipment Costs. Equipment costs are the charges for dedicated equipment used exclusively by an individual user-library. This includes the video display terminal, a data set (which converts the signals between the terminal and the telephone lines), and leased telephone lines. The charges for all these items are fixed by the manufactures and the telephone company. For the BALLOTS-MARC module it is assumed that each library will have one terminal.

(d) Overhead Costs. Overhead costs are the sizeable fixed necessary costs that do not vary directly with the amount of usage of the system. They include the processing cost of major file updates, such as adding new MARC records and deleting old MARC records each week, monthly charges connected with shared file storage devices, and hardware connection time for the use of a small computer to handle communications between the terminals and the 360/67. Update processing and utilities costs are based on the CPU rate as are on-line and batch transactions; but since this processing is measured in minutes per day and hours per week, it amounts to much more than the small transaction costs. The charges for storage devices are based on manufactures' rental rates. The overhead costs are distributed among user-libraries according to the volume of batch and on-line transactions for each library relative to the volume for all users.

4. Performing the Manual-Automated Cost Study

With the BALLOTS-MARC function flow charts in hand, next determine the amount of time spent by professionals, assistants, and students in accomplishing each activity. The same methodology is employed that was used for calculating manual personnel costs.

(a) Remaining Manual Costs. Some manual activities, including the amounts of material they process, will be unaffected by automation. Therefore the personnel times will remain the same, as will, presumably, the types of personnel involved. In addition, what was previously a single manual activity may now be divided into two parts: a manual part that processes some portion of the original activity's work, plus a manual-automated part (at the CRT terminal) that handles the remainder of the work. Or, an entirely new manual-automated activity may be required. The personnel times for manual activities thus affected must be recalculated. In many cases, it is sufficient to reduce personnel time in proportion to the reduction in the amount of work to be handled. For example, consider a manual purchase order typing activity that may produce 10,000 purchase orders and that requires 900 hours of assistant time and 100 hours of professional time. If half, or 5,000, of the purchase orders will be computer-printed in the BALLOTS-MARC system, then the remaining 5,000 purchase orders can be produced in the remaining manual typing activity in 450 hours of assistant time and 50 hours of professional time. Keep in mind that the portion of work left to be processed manually may be the more difficult work. If this is the case, add some time onto that allotted to the remaining manual activity. The new manual-automated activity that would replace the previous manual typing activity is keying ordering data at a CRT terminal. The following section discusses the calculation of personnel times for such new activities.

(b) New Manual-Automated Activities. In order to calculate personnel times, standard times are used for all activities performed at a CRT terminal. One or more of the following activities are performed in each function for which manual-automated costs are to be calculated.

PROCEDURE (using CRT terminal)	STANDARD TIME PER UNIT
Search MARC	1 minute

Key ordering data and acquisition bibliographic updates	5 minutes
Key ordering data only	3 minutes
Key holdings data and cataloging bibliographic updates	5 minutes
Key holdings data only	3 minutes
Establish standing search request	2 minutes
Purge standing search request	1 minute

The number of times the activity is performed in the function is multiplied by the standard time for the activity. This gives the productive personnel time for the activity involving CRT terminal use. These standard times were extrapolated from standard times arrived at through a BALLOTS prototype technical processing system run during 1969.

(c) Batch and On-Line Charges. Batch and on-line charges are calculated from the number of transactions used by each function. To calculate these costs, a new form is used--the On-Line/Batch Transaction Summary form. The form plus a filled-out sample are included in Appendix A. This form provides a convenient worksheet to record the number of each kind of transaction used by the function, and shows the high and low unit cost of each for easy calculation. The total high and low charges in part C of the form are carried over to the "Other Costs" figure in part I.B of the Function Summary Form and are part of the calculation of the high and low Total Function Cost.

D. Cost Analysis Summary

In performing the cost analysis a great deal of data is accumulated that must be summarized for easy reference. A series of Library Cost Analysis summary forms have been designed to display pertinent data. These forms are in Appendix D and are described below.

1. COMPARATIVE PERSONNEL HOURS AND COSTS, BY FUNCTION, FOR MANUAL AND BALLOTS-MARC MODULE

This form summarizes the data from all the Function Summary Forms used. It is, as it says, a summary, by function, of personnel hours and costs. It does not include computing overhead, supplies, equipment, or transaction charges. The highlighted box in the lower right-hand corner displays the total plus or minus effect on personnel costs of using the BALLOTS-MARC system.

2. COMPARATIVE TOTAL ANNUAL FUNCTION COSTS

The total function costs for manual functions consist of personnel, supplies, and equipment costs. The total function costs for the functions affected by BALLOTS-MARC include personnel, supplies, equipment, and on-line and batch transaction charges. This form compares the (high) total costs of performing a function manually and with the aid of BALLOTS. The highlighted box in the lower right-hand corner displays the total plus or minus effect on the total function costs of using the BALLOTS-MARC system--excluding the computing overhead and equipment costs.

3. SUMMARY TRANSACTION VOLUMES AND CHARGES, BY FUNCTION

The On-line/Batch Transaction Summary form records the summarized on-line and batch transaction volumes and charges for each function. Both high and low charges are summarized.

4. COMPUTING OVERHEAD AND DEDICATED EQUIPMENT COSTS

Appendix C contains data to calculate the overhead and equipment charges of BALLOTS-MARC to user-libraries. Using the data in that appendix, record the monthly and annual charges on this form. As other sections of this report state, each component of the computing overhead for a library is figured as the ratio of that library's total on-line and batch transaction volume (the sum of columns 2 and 3 in form 3 described above) to the total number of transactions handled by the system for all the network libraries. Use a "best guess" for this form based on the examples in Appendix C, or call Eleanor Montague, Stanford University, Stanford, California, 321-2300, extension 3741.

5. SUMMARY IMPACT OF USING BALLOTS-MARC

This form summarizes data from the four summary forms just described. Part 1 lists the total annual payment to Stanford and to terminal manufacturers for participation in the BALLOTS network. The second part subtracts anticipated personnel savings (and any equipment and supplies savings) from this total payment to yield the EFFECTIVE ANNUAL COST TO A USER LIBRARY. (Of course, if personnel or equipment costs are increased, the amount of increase is added to part 1.) Keep in mind that unless these personnel savings are actually realized, the effective annual cost equals the Total Annual Payment of part 1.

IV IMPACT OF NETWORK PARTICIPATION

The original four-library feasibility study was undertaken because all agreed that management had to have reliable cost data in order to decide whether to postpone automation, join a network, or undertake independent library automation. The data gathered during the study gave a complete picture of the cost impact to each library participating in one particular network--the Stanford BALLOTS network. In addition to costs, each library gained other information valuable for management decision, about such things as: high manual-cost areas; complicated procedures that could be simplified; activities being performed by the wrong category of personnel; like activities being performed in different functions; etc. At the end of the study, each library was able to summarize the non-economic advantages and disadvantages of using the BALLOTS-MARC system.

The following sections summarize and highlight the findings and conclusions of the feasibility report, and, where pertinent, the Stanford Cost Study.

A. Summary Statistics

(Stanford cost figures are included in all ranges. One of the original four feasibility study libraries discovered errors in its cost calculations and is therefore not included.) For each library, the personnel cost comparison between performing functions manually and performing the same functions with the aid of BALLOTS-MARC showed a saving in personnel costs, since computer processing would replace some manual processing. Including Stanford, the annual personnel savings ranged from $2,000 to $18,900. However, computer transaction charges were an added cost factor, ranging from $800 to $20,600. In the case of one library, the personnel savings were almost three times the amount of transaction charges. For the rest of the libraries, however, the cost of the batch and on-line transaction charges nearly equalled the personnel savings. The number of purchase orders to be computer printed by the system ranged from 400 to 9,000; and the number of catalog cards ranged from 3,000 to 113,000.

Computing overhead and dedicated equipment charges for Stanford plus the three other libraries ranged from a low of $10,000 to a high of $38,400. The net annual cost to each library for participating in BALLOTS-MARC (the sum of computing overhead, dedicated equipment, and transaction charges, minus personnel savings) ranged from $8,660 to $38,655. The effect on overhead costs of more libraries' joining the network is shown in Appendix C.

The charges for equipment, on-line and batch transaction and computing overhead make up the amount paid each year in dollars by each library to the Stanford Computation Center, to the telephone company, and to equipment manufacturers. In general, this is an added cost to the library and represents added cash flow. The difference between personnel costs under the current manual system and personnel costs under the manual-automated system is a function cost differential that is, in effect, a saving to the library. But personnel savings will not appear in the cash flow unless there is a net reduction in staff. Reassigning staff to new work is a benefit in terms of the added service performed by the library, but not in terms of cash flow savings.

B. Summary Advantages and Disadvantages

In Volume 1 of the feasibility report, each participating library discussed the advantages and disadvantages of its participating in the BALLOTS network. Below is a composite summary of all their remarks.

Each library felt that an on-line system available on demand to technical processing staff, based on a file of Library of Congress bibliographic records with multiple indices, was an advantage that would speed searching and processing; but all were concerned that a six-month MARC file (assumed for the cost calculations) would not be as useful as a file covering a longer period. The standing search feature was felt to be a strong addition to the module that would increase the usefulness of the system. The limited scope of the BALLOTS-MARC module was attractive because it would not disrupt ongoing operations or set the participating libraries on an irreversible course towards automation. The modular approach was well received because changes in library environment and personnel functions would thus be gradual. Most of the libraries felt that the use of the BALLOTS-MARC module would stimulate staff morale and that current staffing levels would be sufficient, with training, to perform the new manual-automated activities. Presorted computer-printed outputs and operating statistics were also seen to be advantages.

It was felt that the procedure of guessing which of the books being processed could be found in MARC would be a problem that hopefully could be solved with experience. Since the computer-printed cards would have a different format than standard cards, there was concern over the public reaction to the mixture in the catalog. The scope of the MARC file and the module is such that existing expensive manual card files of proofslips, etc., will still have to be maintained. The libraries hoped that in the future these files could be discontinued.

REFERENCES

1. Eleanor Montague, Douglas Ferguson, et al., FEASIBILITY STUDY ON THE PARTICIPATION OF FOUR COLLEGES AND UNIVERSITIES IN A STANFORD UNIVERSITY LIBRARY AUTOMATION NETWORK, SPIRES/BALLOTS Project, Stanford University, Stanford, California, July 1971.

2. Originally, it was a study objective to derive costs that would permit comparison between libraries of the costs and times taken to produce a given unit. As the study progressed, however, this interlibrary comparison requirement interfered with the prime objective, which was to produce costs for a library that compared the manual system with the same system supported by BALLOTS-MARC. The two objectives clashed primarily in the definition and scope of various functions. But even though a strict comparison between libraries at the function level is sometimes difficult, activities (being smaller) can be compared for standard times, category of personnel involved, etc.

3. The feasibility workshop was begun without a predetermined method for doing the cost analysis. A review of the literature on the subject indicated that a variety of methods have been developed under widely varying circumstances of library size, study objectives, validity requirements, and documentation to be produced. It was the purpose of the original workshop study to achieve some fairly specific objectives (see I.B) in a period of 13 weeks. An equally important intention was to encourage the study team to involve itself fully in defining the study methodology and to benefit from the critical comments and experience of all team members. The resulting costing methodology used in the study was the product of a joint effort by the entire feasibility study team.

After the report was issued, the methodology was applied to a cost analysis of the Stanford University Libraries technical processing system. This study was done by a BALLOTS analyst and a senior librarian from Stanford's Meyer Undergraduate Library. During the Stanford study, the cost methodology used in the feasibility study was subjected to new factors; the result was the development of alternative methods, refinements, and extensions. The methodology described in this summary, then, is a combination of that used by the study team and that used in the Stanford cost analysis.

4. Special business or cost training is not mandatory for this analysis. However, familiarity with technical processing activities and work flow is required. If the library does have a financial manager or someone in a similar position, perhaps help can be obtained from that source.

5. There is certainly more than one way to calculate costs in an analysis designed for a purpose like ours. We suggest using the adjusted hourly wage approach, with the actual number of productive hours required to perform an activity, because such an approach yields function costs that are more nearly the true cost to the library of performing that function. This is true because the amount of money the library must pay for nonproductive time is allocated across the rate paid for every hour spent in productive activity. If this is an unnatural or very difficult approach to use for a particular library, other methods can be used. Caution, however, is suggested in making sure that (1) the underlying assumptions for the methodology are stated clearly, (2) the costs derived for manual functions and those for manual-automated functions are comparable, and (3) when function costs and personnel costs are being compared with the costs from other libraries that have performed this study, the differences in methodology, and therefore the apparent differences in costs, are clearly understood.

6. The methodology presented in this section for calculating average annual salary is simple and therefore crude. If more precision is desired, weight the average of each classification according to the number of people in that classification when calculating the overall average for a category of personnel.

7. Some libraries found it helpful at this point to look ahead to data required for calculating manual-automated costs--namely, the statistics questionnaire and schematic diagram described in III.C.2, with examples given in Appendix B.

APPENDIX A

DATA COLLECTION FORMS

Function Worksheet

Function Worksheet Instructions

Filled-out Function Worksheet

Function Summary Form

Function Summary Form Instructions

Filled-out Function Summary Form

On-line/Batch Transaction Summary Form

Filled-out Transaction Summary Form

FUNCTION WORKSHEET	1 Library:	Page: of
	Function:	Date:
	Author(s):	Status:

2 ACTIVITY NAME		3 PERSONNEL COSTS						
		A Category	B No.	C Total hrs/wk	D Wks/yr	E Hrs/yr	F Adj. rate/hr	G Adj. yr cost
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	

4 Comments/Footnotes:

FUNCTION WORKSHEET

Instructions

Block 1 Heading Information

Library: Library name.

Function Name: Standard name of the function described on the worksheet.

Author(s): Initials of the author.

Page: Page___ of ___.

Date: Date the worksheet was completed. If revised, put "rev." after the date given.

Status: 1st draft, 2nd draft, Final, etc.

Block 2 Activity Name

Use standard activity name. If there is enough space, include a brief scope description of the activity. If necessary, attach a separate page to explain the scope of the activity.

Block 3 Personnel Costs

Data for each activity is subdivided by category of personnel: line 1 for Professional (P), line 2 for Assistants (A), and line 3 for Students (S). Line 4 (TOTAL) sums up activity data for all categories of personnel.

Use lines 1, 2, and 3 as needed, using the following instructions for each column. Leave unused lines blank.

Col. A Category of Personnel
P - Professional
A - Assistant
S - Student
TOTAL - Activity total

Supply the following information (cols. B-G) for each category of personnel.

Col. B Number of Persons
The number of persons engaged in the activity. This information is not required for cost calculations; supply only if known and of interest.

Col. C Total Hours Per Week
The total number of hours per week spent by all persons engaged in the activity. Represent fractions of hours as decimal numbers. Two decimal places is sufficient detail.

Col. D Weeks Per Year
The number of weeks per year that this activity is pursued. For example, an ongoing activity is pursued 52 weeks per year, whereas a seasonal activity might require only 12 weeks per year. Col. D should reflect the number of weeks per year this activity is performed. With rare exception (e.g., special summer projects), all activities will need 52 weeks.

Col. E Hours Per Year
Multiply col. C by col. D and enter the result in col. E.

Col. F Adjusted Rate Per Hour
The standard rate per hour for that category of personnel, adjusted for overhead, vacation, breaks, etc. Use this same rate for personnel category for all activities in a function.

Col. G Adjusted Yearly Cost
Multiply col. E by col. F and enter the result in col. G.

Block 4 Comments/Footnotes

Use this space for comments or to footnote any information on the worksheet, to explain a special situation, etc. It is a wise practice to footnote liberally. This makes it easier to audit your own work and for any other person reading the form.

FUNCTION WORKSHEET	1 Library: H	Page: 1 of 1
	Function: Distribution	Date: 6/15/71
	Author(s): LOM	Status: FINAL

2 ACTIVITY NAME		3 PERSONNEL COSTS A Category	B No.	C Total hrs/wk	D Wks/ yr	E Hrs/ yr	F Adj. rate/hr	G Adj. yr cost
Manual search	1	P	1	10	52	520	6.62	3442.40
	2	A	2	30	52	1,560	4.53	7066.80
	3	S						
	4	Total	---	40	---	2,080	---	10509.20
MARC search	1	P	1	5.0	52	260	6.62	1721.00
	2	A	1	2.1	52	109.2	4.53	494.68
	3	S						
	4	Total	---	7.1	---	369.2	---	2215.88
Est. SSR	1	P						
	2	A	1	.7	52	36.4	4.53	164.89
	3	S						
	4	Total	---	.7	---	36.4	---	164.89
Match MARC with books	1	P						
	2	A	1	2	52	104	4.53	471.12
	3	S	1	2	52	104	2.05	213.20
	4	Total	---	4	---	208	---	684.32
Pull books on the purged list to be searched manually	1	P						
	2	A	1	2	52	104	4.53	471.12
	3	S						
	4	Total	---	2	---	104	---	471.12
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	
	1	P						
	2	A						
	3	S						
	4	Total	---		---		---	

4 Comments/Footnotes:

MARC search (SI01 + BF01) @ 1 minute 6,419
Estimated SSR (SF01) @ 1 minute 2,140

FUNCTION SUMMARY	1 Library:	Page ___ of ___
	Function Name:	Date:
	Author(s):	Status:

2

I. COSTS

A. Personnel Costs (Adjusted yearly cost)

Professional . . . $________

Assistant. . . . . $________

Student. . . . . . $________

Total. . . . . $________

B. Other Costs

Equipment. . . . . $________

Supplies . . . . . $________

Other. . . . . . . $________(high)

$______(low)

Total. . . . . $________(high)

$______(low)

C. TOTAL FUNCTION COST . . . . $________(high)

$______(low)

3

II. PERSONNEL TIME (HRS/WK)

A. Professional . . . . . . .________hrs/wk

B. Assistant. . . . . .________hrs/wk

C. Student. . . . . .________hrs/wk

D. TOTAL. . . . . .________hrs/wk

4

III. PERSONNEL TIME (HRS/YR)

A. Professional . . . . . . .________hrs/yr

B. Assistant. . . . . .________hrs/yr

C. Student. . . . . .________hrs/yr

D. TOTAL. . . . . .________hrs/yr

5

IV. PRODUCTION - UNITS/YEAR

6

V. UNIT COST

7

VI. COMMENTS AND CALCULATIONS:

FUNCTION SUMMARY FORM
Instruction Sheet

Block 1 Heading Information

See Block 1 information on FUNCTION WORKSHEET Instruction Sheet.

Block 2 Costs

A. Personnel Costs (Adjusted Yearly Cost)

Professional: the sum of line 1, col. G figure for all activities on the Function Worksheet.

Assistant: the sum of line 2, col. G figures for all activities on the Function Worksheet.

Student: the sum of line 3, col. G figures for all activities on the Function Worksheet.

Total: the sum of the three personnel cost figures above. As a double check, the total sum here should equal the sum of line 4, col. G figures for all activities on the Function Worksheet.

B. Other Costs

Equipment: annual equipment cost for the function, figured according to established conventions. If no equipment is used, leave the space blank.

Supplies: the total yearly cost of supplies used in the function, calculated according to established assumptions.

Other: any other miscellaneous costs incurred by the function. For manual-automated procedures, record here the high and low totals from the On-line/Batch Transaction Summary form, C1 and C2.

Total: the total cost (for the function) for equipment, supplies, and miscellaneous costs. Show both the high and low totals.

C. Total Cost

The sum of A. Total and B. Total (Function Summary Form). Show both the high and low totals.

Block 3 Personnel Time (Hrs/Wk)

A. Professional

The sum of line 1, col. C figures for all activities on the Function Worksheet.

B. Assistant

The sum of line 2, col. C figures for all activities on the Function Worksheet.

C. Student

The sum of line 3, col. C figures for all activities on the Function Worksheet.

D. Total

The sum of figures in A., B., and C., Block 3 (Function Summary Form). This sum should equal the sum of line 4, col. C figures for all activities on the Function Worksheet.

Block 4 Personnel Time (Hrs/Yr)

The sum of line 1, col. E figures for all activities on the Function Worksheet.

B. Assistant

The sum of line 2, col. E figures for all activities on the Function Worksheet.

C. Student

The sum of line 3, col. E figures for all activities on the Function Worksheet.

D. Total

The sum of figures in A., B., and C., Block 4. This sum should equal the sum of line 4, col. E figures for all activities on the Function Worksheet.

Block 5 Production - Units/Year

The total production per year of the function being summarized.

Block 6 Unit Cost

Cost per unit is calculated as follows:

<Block 2, Costs, part C., Total Cost figure on Function Summary Form> divided by <Block 5, Production - Units/Year figure on Function Summary Form>= <Block 6, Unit Cost> per year.

Block 7 Comments and Calculations

Any comments pertinent to the description of the function. For example, one would use this space to state any costs or activities excluded. Include also any calculations hidden in cost figures.

--SAMPLE--

FUNCTION SUMMARY	1 Library: H	Page 1 of 1
	Function Name: Distribution	Date: 6/15/71
	Author(s): LOM	Status: FINAL

2

I. COSTS

A. Personnel Costs (Adjusted yearly cost)

Professional . . . $ 5,163.00

Assistant. . . . . $ 8,668.61

Student. . . . . . $ 213.20

Total. . . . . $ 14,045.41

B. Other Costs

Equipment. . . . . $

Supplies . . . . . $

Other. . . . . . . $ 1,379.30 (high)
$ 694.95 (low)

Total. . . . . $ 1,379.30 (high)
$ 694.95 (low)

C. TOTAL FUNCTION COST . . . . $ 15,424.71 (high)
$14740.35 (low)

3

II. PERSONNEL TIME (HRS/WK)

A. Professional . . . . . . 15.0 hrs/wk

B. Assistant. . . . . 36.7 hrs/wk

C. Student. . . . . 2.0 hrs/wk

D. TOTAL. . . . . 53.7 hrs/wk

4

III. PERSONNEL TIME (HRS/YR)

A. Professional . . . . . . 280.0 hrs/yr

B. Assistant. . . . . 1,913.6 hrs/yr

C. Student. . . . . 104.0 hrs/yr

D. TOTAL. . . . . 2,792.4 hrs/yr

5

IV. PRODUCTION - UNITS/YEAR

8,666 titles per year

6

V. UNIT COST

$1.78 (high) } per title
$1.70 (low) }

7

VI. COMMENTS AND CALCULATIONS:

ON-LINE/BATCH TRANSACTION SUMMARY ①	LIBRARY:	PAGE: ____ OF: ____
	FUNCTION:	DATE:
	AUTHOR(S):	STATUS:

A. On-Line Transactions Charges ②

	No.	x Unit Cost	= Total
1. Searches			
a. simple	___	$.04	___
		.02	___
b. medium	___	.17	___
		.08	___
c. complex	___	.33	___
		.17	___
d. Total	___	--	
(high)			___
(low)			___
2. Simple Input/ Update			
a. Key acquisition Data	___	$.10	___
		.05	___
b. Key Bibliographic Update - 1 ③	___	.10	___
		.05	___
c. Key Bibliographic Update - 2 ③	___	.10	___
		.05	___
d. Key Holdings Data	___	.10	___
		.05	___
e. Establish SSR's	___	.10	___
		.05	___
f. Purge SSR's	___	.10	___
		.05	___
g. Total	___	--	
(high)			___
(low)			___

B. Batch Transactions Charges ④

	No.	x Unit Cost	= Total
1. Processing Costs			
a. Purchase Orders	___	$.06	___
		.04	___
b. cards	___	.06	___
		.04	___
c. spine labels	___	.06	___
		.04	___
d. match SSR's	___	.06	___
		.04	___
e. slips	___	.06	___
		.04	___
f. purge SSR's	___	.06	___
		.04	___
g. Total	___	--	
(high)			___
(low)			___

C. Total Batch and On-Line Transactions Charges ⑤

1. high ________
2. low ________

D. Total Batch and On-Line Transaction Volumes ⑥

1. Total ________

①) Detail on Section I. B, "Other Costs," on Function Summary form.
②) Calculate both a high and low unit cost.
③) Key bibliographic update-1 refers to changing data from main entry through collation. Key bibliographic update-2 refers to changing data from notes through tracings.
④) The high and low unit costs here include a printer charge of $.02 per individual printed output.
⑤ High and low totals from A. 1. d., A. 2. g, and B. 1. g.
⑥ Totals from A. 1. d, A. 2. g, and B. 1. g.

ON-LINE/BATCH TRANSACTION SUMMARY ①	LIBRARY: H	PAGE: 1 OF: 1
	FUNCTION: Distribution	DATE: 6/15/71
	AUTHOR(S): LOM	STATUS: FINAL

A. On-Line Transactions Charges ②

	No. x	Unit Cost =	Total
1. Searches			
a. simple	1,925.	$.04	77.00
		.02	38.50
b. medium	3,851.	.17	654.67
		.08	308.08
c. complex	641.	.33	211.53
		.17	108.97
d. Total	6,417.	--	
(high)			943.20
(low)			455.50
2. Simple Input/ Update			
a. Key acquisition Data	___	$.10	___
		.05	___
b. Key Bibliographic Update - 1 ③	___	.10	___
		.05	___
c. Key Bibliographic Update - 2 ③	___	.10	___
		.05	___
d. Key Holdings Data	___	.10	___
		.05	___
e. Establish SSR's	2,140.	.10	214.00
		.05	107.00
f. Purge SSR's	937.	.10	93.70
		.05	46.85
g. Total	3,077.	--	
(high)			307.70
(low)			153.85

B. Batch Transactions Charges ④

	No. x	Unit Cost =	Total
1. Processing Costs			
a. Purchase Orders	___	$.06	___
		.04	___
b. cards	___	.06	___
		.04	___
c. spine labels	___	.06	___
		.04	___
d. match SSR's	1,203	.06	72.18
		.04	48.12
e. slips	___	.06	___
		.04	___
f. purge SSR's	937	.06	56.22
		.04	37.48
g. Total	2,140	--	
(high)			128.40
(low)			85.60

C. Total Batch and On-Line Transactions Charges ⑤

1. high 1,379.30
2. low 694.95

D. Total Batch and On-Line Transaction Volumes ⑥

1. Total 11,634

① Detail on Section I. B, "Other Costs," on Function Summary form.

② Calculate both a high and low unit cost.

③ Key bibliographic update-1 refers to changing data from main entry through collation. Key bibliographic update-2 refers to changing data from notes through tracings.

④ The high and low unit costs here include a printer charge of $.02 per individual printed output.

⑤ High and low totals from A. 1. d., A. 2. g, and B. 1. g.

⑥ Totals from A. 1. d, A. 2. g, and B. 1. g.

APPENDIX B

TRANSACTION VOLUMES

Transaction Estimate Questionnaire

Filled-out Transaction Estimate Questionnaire

BALLOTS-MARC System Schematic (Keyed to the Transaction Questionnaire)

BALLOTS-MARC TRANSACTION ESTIMATE QUESTIONNAIRE

MARC File Assumption: ( )6-month; ( )12-month; ( )English Lang.; ( )Roman Alpha

Library:____________________ Date:____________________

Author:____________________

All figures are annual - Questions are keyed to BALLOTS-MARC Schematic

A. For book material acquired:

% English Lanugage	________	% English
% Non-English, roman alphabet	________	% Non-English, roman
% Other (Non-roman alphabet)	________	% Non-roman

B. For book material acquired:

% Current (imprint date within last two years)	________	% Current
% Non-current	________	% Non-Current

1. Number of PO's currently typed ________ Current PO's

2. Number of titles to be ordered (new or added) for which MARC is likely. (This number is equal to the number of MARC searches prior to ordering.) ________ Titles

3. Number of titles to be ordered (new or added) for which MARC is <u>not</u> likely. ________ Titles

4. Number of titles for which MARC will be found prior to ordering:

   a. New titles ________ Titles
   b. Added copies ________ Titles
   c. Total MARC found ________ Titles

   (Note: This equals the number of Simple acquisition input screens to be keyed.)

5. Number of titles searched in MARC but not found (no. 2 minus no. 4c.) ________ Titles

6. Number of PO's manually typed for:

   a. MARC data not likely ________ PO's
   b. MARC searched, not found ________ PO's
   c. Total manually produced PO's ________ PO's

7. Number of PO's produced from MARC (equals sum of nos. 4a and 4b) ______________ MARC PO's

8. Number of 3"x5" slips per title required to maintain manual files ______________ Slips/title

9. Total number of computer-produced 3"x5" slips per year (equals no. 8 times no. 7) ______________ Total slips

10. Total number of volumes received (PO and NPO) ______________ Volumes

11. Material received, MARC not likely:

    a. Number of volumes ______________ Volumes
    b. Number of titles ______________ Titles

12. MARC PO material received: (assume all titles ordered are received)

    a. Number of volumes ______________ Volumes
    b. Number of titles ______________ Titles

13. Material received, MARC likely (non-MARC PO material and NPO material):

    a. Number of volumes ______________ Volumes
    b. Number of titles ______________ Titles

14. Material received, MARC likely:

    a. MARC found after search
        1. Number of volumes ______________ Volumes
        2. Number of titles ______________ Titles

    b. MARC not found
        1. Number of titles held for MARC ______________ Titles
        2. Number of titles not held for MARC ______________ Titles
        3. Number of volumes not held ______________ Volumes

15. Number of standing search requests established (should equal figure in no. 14b-1) ______________ SSR's

16. Computer-produced catalog cards:

    a. Number of SETS produced when MARC searched and found after receipt (should equal figure in no. 14a-2) ______________ MARC card sets

    b. Number of sets produced when MARC found prior to ordering (should equal figure in no. 12b) ______________ MARC card sets

    c. Number of sets produced when MARC found by SSR (equals no. 24b) ______________ MARC card sets

    d. Total sets * ______________ MARC card sets

17. MARC spine labels:

    a. Number of labels produced when MARC searched and found after receipt (should equal figure in no. 14a-1) ______________ MARC spine labels

    b. Number of labels produced when MARC found prior to ordering (should equal figure in no. 12a) ______________ MARC spine labels

    c. Number of labels produced when SSR match found (equals figure in 24a) ______________ MARC spine labels

    d. Total labels ______________ MARC spine labels

18. Average number of cards in a card set ______________ Av. cards/set

19. Number of spine labels per volume (e.g., SUL produces two for each volume) ______________ Spine labels/volume

20. Number of cards currently produced manually ______________ Cards

*Note: This equals the number of Simple input-Holdings Screens to be keyed.

21. Number of volumes currently processed in End Processing ______________ Volumes

22. Number of volumes going to End Processing with MARC data found (equals sum of nos. 14a-1, 12a, and 24a) ______________ Volumes

23. Number of volumes going to End Processing with no MARC data found (no. 21 minus no. 22 should equal no. 23) ______________ Volumes

24. Material with MARC records caught during SSR matching (this figure indicates annual figure for all material matched by an SSR during holding period):

    a. Volumes ______________ Volumes
    b. Titles ______________ Titles

    (Note: 24b equals the number of Simple input SSR purges to be keyed.)

25. Total number of MARC catalog cards (equals figures in no. 16 times figure in no. 18) ______________ MARC cards

26. Total number of MARC spine labels (equals figure in no. 17d times figure in no. 19) ______________ MARC labels

27. Material searched with SSR but not found:

    a. Volumes ______________ Volumes
    b. Titles ______________ Titles

28. Number of cards left to be produced manually if MARC system were operating (equals no. 20 minus no. 25) ______________ Manual cards

10/71

BALLOTS-MARC TRANSACTION ESTIMATE QUESTIONNAIRE

MARC File Assumption: (xx)6-month; ( )12-month; (X)English Lang.; ( )Roman Alph

Library: H Date: 6/18/71
Author: LOM

All figures are annual - Questions are keyed to BALLOTS-MARC Schematic

A. For book material acquired:

% English Lanugage	70	% English
% Non-English, roman alphabet	24	% Non-English, roman
% Other (Non-roman alphabet)	6	% Non-roman

B. For book material acquired:

% Current (imprint date within last two years)	70	% Current
% Non-current	30	% Non-Current

1. Number of PO's currently typed — 10,700 Current PO's

2. Number of titles to be ordered (new or added) for which MARC is likely. (This number is equal to the number of MARC searches prior to ordering.) — 4,280 Titles

3. Number of titles to be ordered (new or added) for which MARC is <u>not</u> likely. — 6,420 Titles

4. Number of titles for which MARC will be found prior to ordering:

a. New titles	1,070	Titles
b. Added copies	0	Titles
c. Total MARC found	1,070	Titles

(Note: This equals the number of Sample acquisition input screens to be keyed.)

5. Number of titles searched in MARC but not found (no. 2 minus no. 4c.) — 3,210 Titles

6. Number of PO's manually typed for:

a. MARC data not likely	6,420	PO's
b. MARC searched, not found	3,210	PO's
c. Total manually produced PO's	9,630	PO's

7. Number of PO's produced from MARC (equals sum of nos. 4a and 4b)	1,070	MARC PO's
8. Number of 3"x5" slips per title required to maintain manual files	6	Slips/title
9. Total number of computer-produced 3"x5" slips per year (equals no. 8 times no. 7)	6,420	Total slips
10. Total number of volumes received (PO and NPO)	22,468	Volumes
11. Material received, MARC not likely:		
a. Number of volumes	13,106	Volumes
b. Number of titles	11,234	Titles
12. MARC PO material received: (assume all titles ordered are received)		
a. Number of volumes	1,176	Volumes
b. Number of titles	1,070	Titles
13. Material received, MARC likely (non-MARC PO material and NPO material):		
a. Number of volumes	7,490	Volumes
b. Number of titles	6,419	Titles
14. Material received, MARC likely:		
a. MARC found after search		
1. Number of volumes	4,991	Volumes
2. Number of titles	4,280	Titles
b. MARC not found		
1. Number of titles held for MARC	2,140	Titles
2. Number of titles not held for MARC	0	Titles
3. Number of volumes not held	0	Volumes
15. Number of standing search requests established (should equal figure in no. 14b-1)	2,140	SSR's

16. Computer-produced catalog cards:

a. Number of SETS produced when MARC searched and found after receipt (should equal figure in no. 14a-2)	4,280	MARC card sets
b. Number of sets produced when MARC found prior to ordering (should equal figure in no. 12b)	1,070	MARC card sets
c. Number of sets produced when MARC found by SSR (equals no. 24b)	1,263	MARC card sets
d. Total sets *	6,553	MARC card sets

17. MARC spine labels:

a. Number of labels produced when MARC searched and found after receipt (should equal figure in no. 14a-1)	4,991	MARC spine labels
b. Number of labels produced when MARC found prior to ordering (should equal figure in no. 12a)	1,176	MARC spine labels
c. Number of labels produced when SSR match found (equals figure in 24a)	1,407	MARC spine labels
d. Total labels	7,574	MARC spine labels

18. Average number of cards in a card set	6	Av. cards/set
19. Number of spine labels per volume (e.g., SUL produces two for each volume)	1	Spine labels/ volume
20. Number of cards currently produced manually	130,384	Cards

*Note: This equals the number of Simple Input-Holdings Screens to be keyed.

21. Number of volumes currently processed in End Processing — 22,468 Volumes

22. Number of volumes going to End Processing with MARC data found (equals sum of nos. 14a-1, 12a, and 24a) — 7,574 Volumes

23. Number of volumes going to End Processing with no MARC data found (no. 21 minus no. 22 should equal no. 23) — 14,894 Volumes

24. Material with MARC records caught during SSR matching (this figure indicates annual figure for all material matched by an SSR during holding period):

    a. Volumes — 1,407 Volumes
    b. Titles — 1,203 Titles

(Note: 24b equals the number of Simple input SSR purges to be keyed.)

25. Total number of MARC catalog cards (equals figures in no. 16 times figure in no. 18) — 39,318 MARC cards

26. Total number of MARC spine labels (equals figure in no. 17d times figure in no. 19) — 7,574 MARC labels

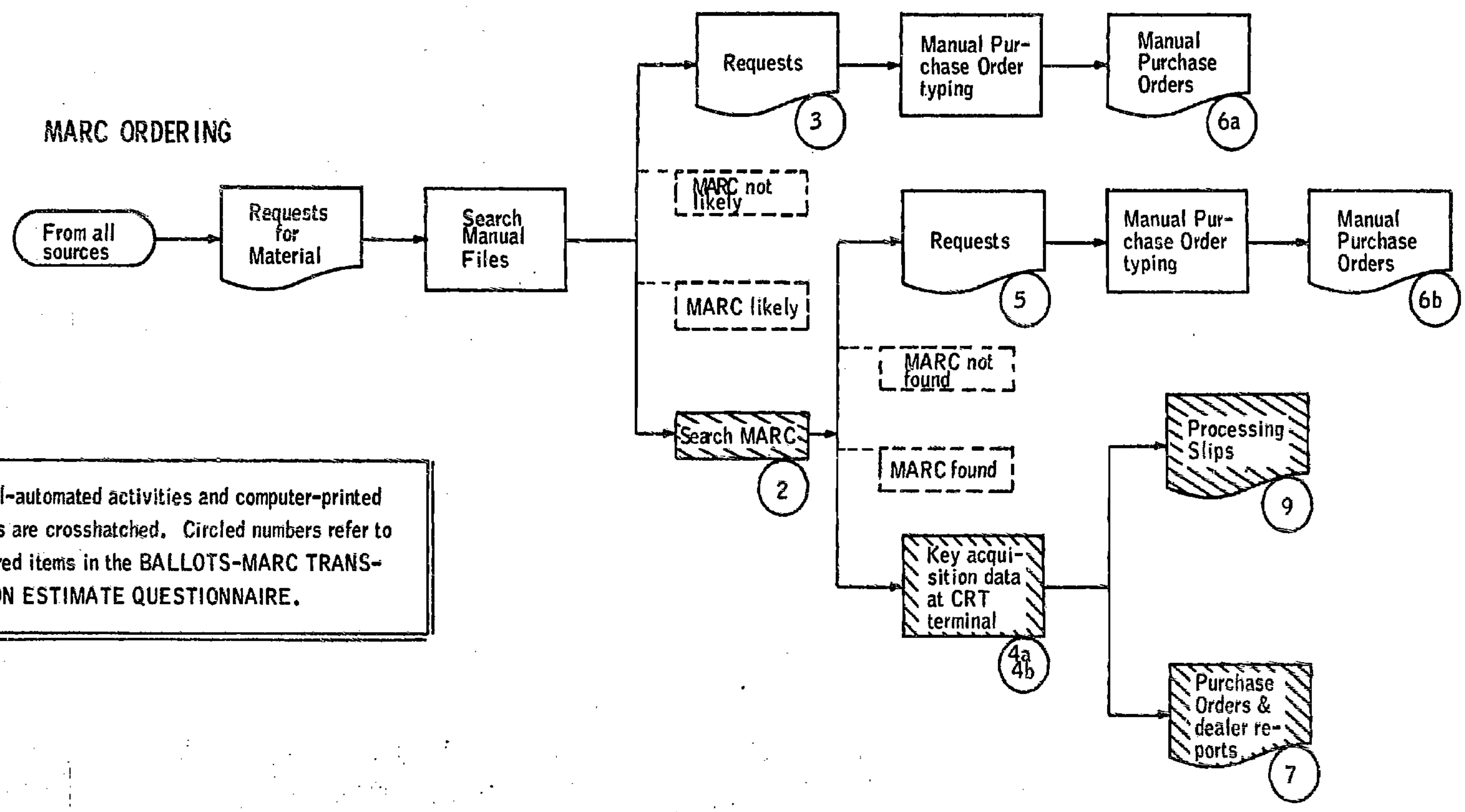
27. Material searched with SSR but not found:

    a. Volumes — 622 Volumes
    b. Titles — 315 Titles

28. Number of cards left to be produced manually if MARC system were operating (equals no. 20 minus no. 25) — 91,066 Manual cards

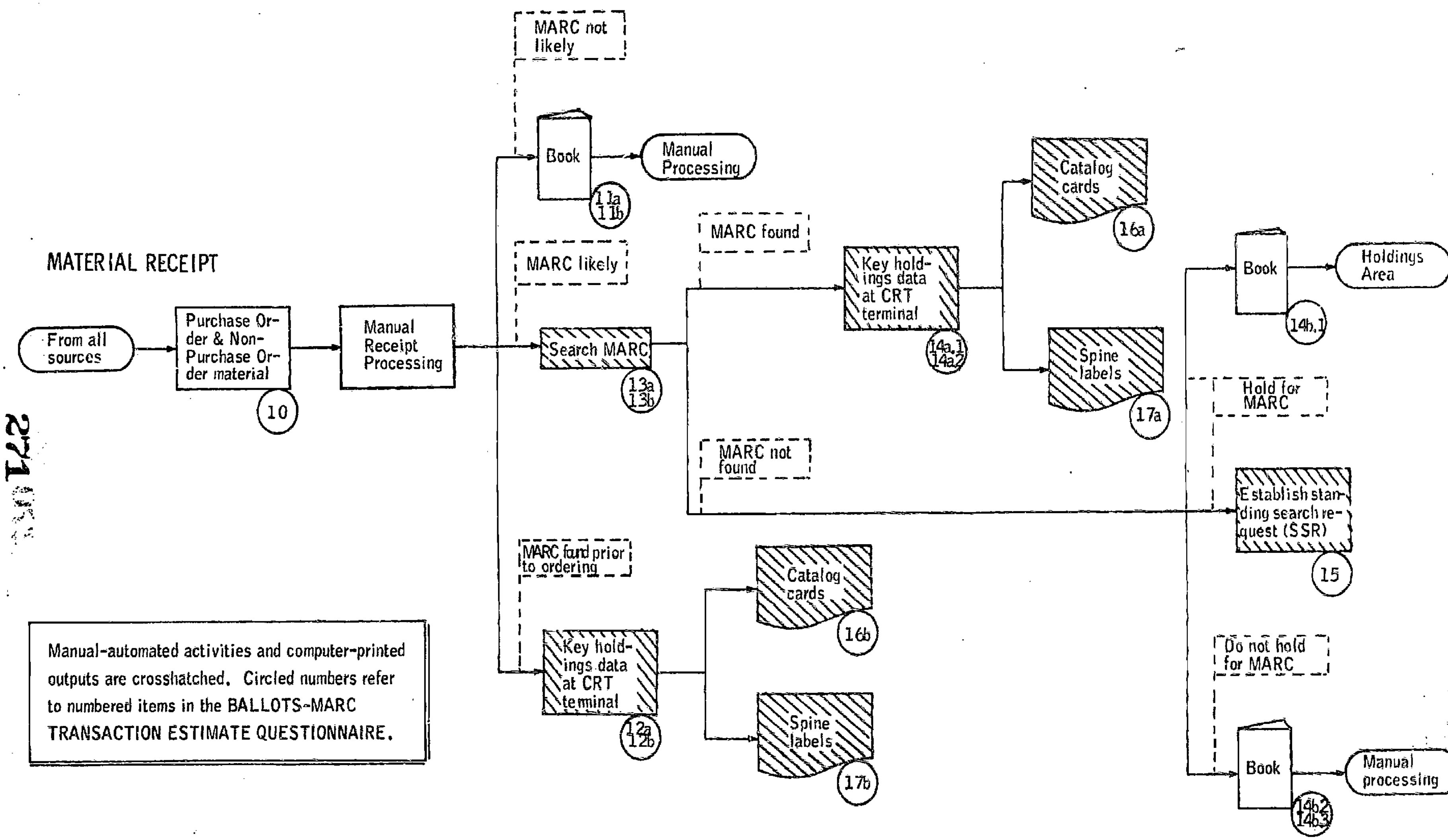
BALLOTS-MARC SYSTEM SCHEMATIC

MARC ORDERING



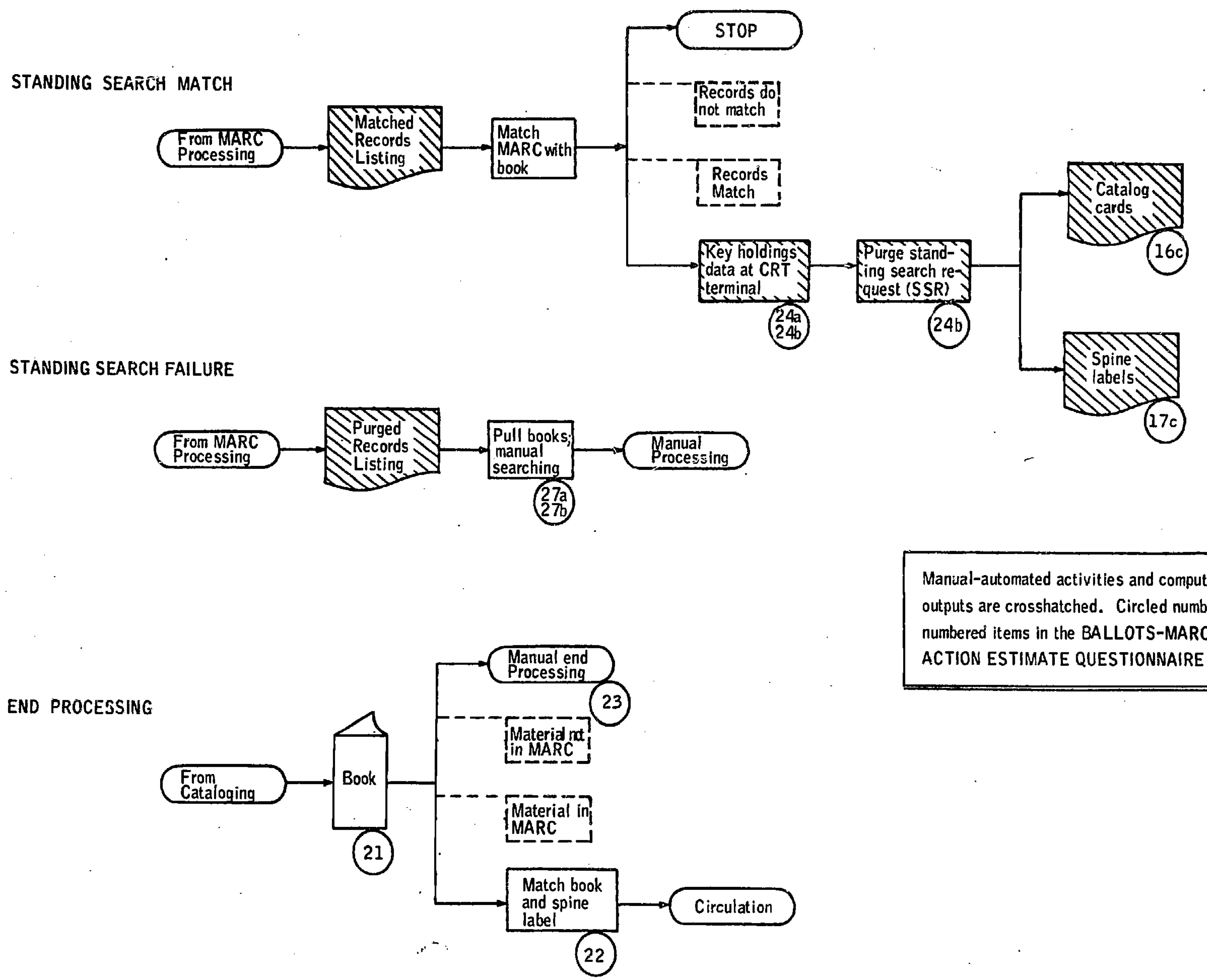
Manual-automated activities and computer-printed outputs are crosshatched. Circled numbers refer to numbered items in the BALLOTS-MARC TRANSACTION ESTIMATE QUESTIONNAIRE.

BALLOTS-MARC SYSTEM SCHEMATIC



Manual-automated activities and computer-printed outputs are crosshatched. Circled numbers refer to numbered items in the BALLOTS-MARC TRANSACTION ESTIMATE QUESTIONNAIRE.

BALLOTS-MARC SYSTEM SCHEMATIC



Manual-automated activities and computer-printed outputs are crosshatched. Circled numbers refer to numbered items in the BALLOTS-MARC TRANSACTION ESTIMATE QUESTIONNAIRE.

APPENDIX C

BALLOTS-MARC CHARGES TO USER LIBRARIES

BALLOTS-MARC CHARGES TO USER LIBRARIES

A. UNIT COSTS FOR TRANSACTIONS

Charged to each school on the basis of individual total monthly transactions.

B. ON-LINE TRANSACTIONS

A high unit cost and a low unit cost are given.

Transaction Type	High	Low
(1) Simple search	$0.04	$0.02
(2) Medium search	$0.17	$0.08
(3) Complex search	$0.33	$0.17
(4) Simple input	$0.10	$0.05

C. BATCH TRANSACTIONS

Printed outputs (1)-(6) include $0.02 print charges.

Transaction Type	High	Low
(1) No. of PO's	$0.06	$0.04
(2) No. of cards	$0.06	$0.04
(3) No. of forms	$0.06	$0.04
(4) No. of labels	$0.06	$0.04
(5) No. of matched SSR's*	$0.06	$0.04
(6) No. of purges SSR's	$0.06	$0.04

D. DEDICATED EQUIPMENT CHARGES (MONTHLY)

(1) Terminal rental	lease $225. (ea.)
(2) Data set rental (one per terminal)	lease $ 50. (ea.)
(3) Leased telephone lines (one per terminal)	**

* Standing search requests.

** Libraries on the Stanford campus do not pay telephone charges, since lines have been purchased. Other libraries should use the following general guide for estimating their average line charges. For the specific charges to your library, contact Eleanor Montague, 321-2300, Extension 3741.

San Jose	- $90.
Oakland	- $130.
San Francisco	- $145.

E. COMPUTING OVERHEAD CHARGES (MONTHLY)

	Monthly Total	(independent of no. of libraries)
(1) Batch utilities and updating	$4,000.	
(2) File storage (one double-density pack)*	$1,600.	
(3) Terminal connection	$ 650.	
	$6,250	

The method of allocating computing overhead costs is to charge each user-library according to its use of the system. For example, if there were four libraries in the network with the following transaction rates, their shares would be as given below.

Example 1:	Library S	Library T	Library U	Library X	TOTAL
No. of trans.	200,000	40,000	60,000	100,000	400,000
% of total	50%	10%	15%	25%	100%
Monthly charges:					
Batch util.	$2,000	$400	$600	$1,000	$4,000
File Storage	$800	$160	$240	$400	$1,600
Terminal Connection	$325	$64	$96	$165	$650
Monthly computing overhead	$3,125	$624	$936	$1,565	$6,250

* The storage costs are contingent on the delivery and acceptance of double-density direct-access storage devices.

Example 2:

If the network consisted of the same four libraries plus ten libraries each of whose transactions amounted to 10,000 and five libraries each of whose transactions amounted to 20,000, the share of each library would be as follows.

	Library S	Library T	Library U	Library X	10 Libraries Y	5 Libraries Z	TOTAL
No. of trans.	200,000	40,000	60,000	100,000	100,000	100,000	600,000
% of Total	33.3%	6.7%	10%	16.7%	16.7%	16.6%	100%
(Breakdown of monthly charges for batch utilities, file storage, terminal connection not shown)							
Monthly computing overhead	$2,082	$417	$625	$1,042	$1,042 (10 @ $104)	$1,042 (5 @ $208)	$6,250

APPENDIX D

COST ANALYSIS SUMMARY FORMS

LIBRARY COST ANALYSIS

COMPARATIVE PERSONNEL HOURS AND COSTS, BY FUNCTION, FOR MANUAL AND BALLOTS-MARC MODULE (Excluding Overhead, Supplies, Equipment, and Transaction Charges)

Library:____________________ Date:____________________

FUNCTION NAME	MANUAL		BALLOTS-MARC		TOTAL ± DIFFERENCE	
	1 TOTAL ANNUAL HOURS (a)	2 TOTAL PERSONNEL COST (b)	3 TOTAL ANNUAL HOURS (a)	4 TOTAL PERSONNEL COST (b)	5 HOURS (c)	6 COST (d)
TOTAL						(e)

(a) Function Summary, III.D.
(b) Function Summary, I;A.TOTAL.
(c) Col 5 = Col 3 - Col 1
(d) Col 6 = Col 4 - Col 2

(e) A minus figure indicates personnel savings realized in BALLOTS-MARC.
A plus figure indicates increased personnel costs using BALLOTS-MARC.

--SAMPLE--
LIBRARY COST ANALYSIS

COMPARATIVE PERSONNEL HOURS AND COSTS, BY FUNCTION, FOR MANUAL AND BALLOTS-MARC MODULE
(Excluding Overhead, Supplies, Equipment, and Transaction Charges)

Library: I   Date: 7/1/71

FUNCTION NAME	MANUAL		BALLOTS-MARC		TOTAL ± DIFFERENCE	
	1 TOTAL ANNUAL HOURS (a)	2 TOTAL PERSONNEL COST (b)	3 TOTAL ANNUAL HOURS (a)	4 TOTAL PERSONNEL COST (b)	5 HOURS (c)	6 COST (d)
Ordering	21,255.0	71,727.50	19,577.0	66,123.81	-167.8	-5,603.69
End Processing	3,380.0	8,112.00	3,250.0	7,800.00	- 13.0	- 312.00
Distribution	3,522.0	14,819.69	3,936.0	17,304.61	+414.0	+2,484.92
LC Cataloging	3,083.6	12,130.14	2,132.4	7,565.57	-951.2	-4,564.57
Card Preparation	12,981.5	55,753.61	10,385.2	44,602.86	-2,596.1	-11,150.75
TOTAL	44,222.1	162,542.94	39,280.6	143,396.85	-3,314.3	(e) -19,146.09

Function Summary, III.D.
Function Summary, I;A.TOTAL.
1 5 = Col 3 - Col 1
1 6 = Col 4 - Col 2

(c) A minus figure indicates personnel savings realized in BALLOTS-MARC.
A plus figure indicates increased personnel costs using BALLOTS-MARC.

LIBRARY COST ANALYSIS

COMPARATIVE TOTAL ANNUAL FUNCTION COSTS (Includes personnel, supplies, equipment and batch and on-line transaction charges. Excludes computing overhead and dedicated equipment costs)

Library:________________ Date:________________

Function Name	1 Manual (a)	2 BALLOTS-MARC (a)	3 Difference (+ or -) (b)
TOTAL			(c)

(a) Function Summary, I.C (High Total Cost)
(b) Column 3 = column 2 - column 1
(c) A minus total indicates the amount of total function savings.
A plus total indicates a net increase in cost for BALLOTS-MARC functions over manual functions.

LIBRARY COST ANALYSIS

COMPARATIVE TOTAL ANNUAL FUNCTION COSTS (Includes personnel, supplies, equipment and batch and on-line transaction charges. Excludes computing overhead and dedicated equipment costs)

Library: I

Date: 7/1/71

Function Name	1 Manual [a]	2 BALLOTS-MARC [a]	3 Difference (+ or -) [b]
Ordering	74,688.86	76,524.45	+1,835.59
End Processing	12,955.00	13,808.82	+ 853.82
Distribution	14,819.69	20,777.22	+5,957.53
LC Cataloging	12,130.14	15,486.85	+3,356.71
Card Preparation	55,753.61	44,602.86	-11,150.75
TOTAL	170,347.30	171.200.20	+ 852.90 [c]

(a) Function Summary, I.C (High Total Cost)
(b) Column 3 = column - column 1
(c) A minus total indicates the amount of total function savings. A plus total indicates a net increase in cost for BALLOTS-MARC functions over manual functions.

LIBRARY COST ANALYSIS

SUMMARY TRANSACTION VOLUMES AND CHARGES, BY FUNCTION

Library: ____________________ Date: ____________________

1 Function Name	2 Batch Transactions (a)	3 On-line Transactions (b)	4 High Transaction Charges (c)	5 Low Transaction Charges (d)
TOTAL				

(a) On-Line/Batch Transaction Summary form, B.1.g.
(b) On-Line/Batch Transaction Summary form, sum of A.1.d and A.2.g.
(c) On-Line/Batch Transaction Summary form, C.1.
(d) On-Line/Batch Transaction Summary form, C.2.

--SAMPLE--

LIBRARY COST ANALYSIS

SUMMARY TRANSACTION VOLUMES AND CHARGES, BY FUNCTION

Library: I

Date: 7/1/71

1 Function Name	2 Batch Transactions (a)	3 On-line Transactions (b)	4 High Transaction Charges (c)	5 Low Transaction Charges (d)
Ordering	66,500	25,000	7,760.00	4,520.18
End Processing	23,000	---	1,591.48	990.32
Distribution	6,400	24,100	3,472.61	1,600.65
LC Cataloging	50,000	16,300	7,920.28	5,090.54
Card Preparation	---	---	---	---
TOTAL	145,900	65,400	20,744.37	12,201.69

(a) On-Line/Batch Transaction Summary form, B.1.g.
(b) On-Line/Batch Transaction Summary form, sum of A.1.d and A.2.g.
(c) On-Line/Batch Transaction Summary form, C.1.
(d) On-Line/Batch Transaction Summary form, C.2.

LIBRARY COST ANALYSIS

COMPUTING OVERHEAD AND DEDICATED EQUIPMENT COSTS

Library:________________ Date:__________________

I. COMPUTING OVERHEAD[a]	MONTH	ANNUAL
a. Batch Utilities and Updating	$________	$________
b. File Storage	________	________
c. Terminal Connection	________	________
TOTAL	$________	$________

II. DEDICATED EQUIPMENT[a]	MONTH	ANNUAL
a. Terminal Rental	$________	$________
b. Data Set Rental	________	________
c. Leased Telephone Lines	________	________
TOTAL	$________	$________

[a] See Appendix C for BALLOTS-MARC Charges to User Libraries.

--SAMPLE--

LIBRARY COST ANALYSIS

COMPUTING OVERHEAD AND DEDICATED EQUIPMENT COSTS

Library: I　　　　　　Date: 7/1/71

I. COMPUTING OVERHEAD[a]	MONTH	ANNUAL
a. Batch Utilities and Updating	$2,000.00	$ 24,000.00
b. File Storage	800.00	9,600.00
c. Terminal Connection	325.00	3,900.00
TOTAL	$3,125.00	$ 37,500.00

II. DEDICATED EQUIPMENT[a]	MONTH	ANNUAL
a. Terminal Rental (2)	$ 500.00	$ 6,000.00
b. Data Set Rental	100.00	1,200.00
c. Leased Telephone Lines	180.00	2,160.00
TOTAL	$ 780.00	$ 9,360.00

[a] See Appendix C for BALLOTS-MARC Charges to User Libraries.

LIBRARY COST ANALYSIS

SUMMARY IMPACT OF USING BALLOTS-MARC

Library:__________________ Date:____________________

1. ANNUAL PAYMENT TO STANFORD OR TERMINAL MANUFACTURES FOR PARTICIPATION IN BALLOTS NETWORK:

a. Computing Overhead $______________

b. Equipment Charges $______________

c. Batch and On-Line Charges (high) $______________

TOTAL $______________

2. EFFECTIVE ANNUAL COST TO USER LIBRARY:

ANNUAL PAYMENT (Total from 1 above) $______________

Minus Difference in Personnel Costs (+ or -)(a) $______________

TOTAL $______________

(a) Total cost difference from the table Comparative Personnel Hours and Costs, by Function.

--SAMPLE--

LIBRARY COST ANALYSIS

SUMMARY IMPACT OF USING BALLOTS-MARC

Library: I   Date: 7/1/71

1. ANNUAL PAYMENT TO STANFORD OR TERMINAL MANUFACTURES FOR PARTICIPATION IN BALLOTS NETWORK:

a. Computing Overhead	$ 37,500.00
b. Equipment Charges	$ 9,360.00
c. Batch and On-Line Charges (high)	$ 20,744.37
TOTAL	$ 67,604.37

2. EFFECTIVE ANNUAL COST TO USER LIBRARY:

ANNUAL PAYMENT (Total from 1 above)	$ 67,604.37
Minus Difference in Personnel Costs (+ or -)[a]	$ 19,146.09
TOTAL	$ 48,458.28

(a) Total cost difference from the table Comparative Personnel Hours and Costs, by Function.