NASA CR-
140345

(NASA-CR-140345) AN ANALYSIS OF
APPLICATIONS DEVELOPMENT SYSTEMS FOR
REMOTELY SENSED, MULTISPECTRAL DATA FOR
THE EARTH OBSERVATIONS DIVISION OF THE
(Rice Univ.) 127 p HC $5.75 CSCL 05B G3/43

N75-11447

Unclas
53858

An Analysis of Applications Development Systems for Remotely Sensed, Multispectral Data for the Earth Observations Division of the NASA Lyndon B. Johnson Space Center

by

D. L. Van Rooy, R. M. Smith* and M. S. Lynn
Institute for Computer Services & Applications
Rice University

Prepared for the Earth Observations Division
under Contract NAS 9 -12776
National Aeronautics and Space Administration
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas

February 1974

* Independent Consultant

275-025-018

An Analysis of Applications Development Systems for Remotely Sensed, Multispectral Data for the Earth Observations Division of the NASA Lyndon B. Johnson Space Center

by

D. L. Van Rooy, R. M. Smith*, and M. S. Lynn
Institute for Computer Services & Applications
Rice University

ABSTRACT

This study examines in depth the status of and needs for an application development system (ADS) for remotely sensed, multispectral data at the Earth Observations Division (EOD) at JSC. A "top-down" approach was used whereby fundamental areas (designated design goals) that such an ADS should address are detailed, followed by basic features (designated design objectives) that ideally such a system should contain. The design objectives were then prioritized according to the needs of EOD's program objectives.

Four systems (ERIPS, ASTEP, LARSYS Batch, and LARSYS 3) available to EOD were then measured against the ideal ADS as defined by the design objectives and their associated priorities. This was accomplished by rating each of the systems on each of the design objectives. Utilizing the established priorities, it was then determined how each system "stood up" as an ADS. Recommendations were then made as to possible courses of action for EOD to pursue to obtain a more efficient ADS. These alternative recommendations are offered without any quantitative consideration being given to the operating constraints (e.g. cost of implementation) of EOD, since only EOD itself can determine these.

Institute for Computer Services & Applications
Rice University
Houston, Texas

February, 1974

This analysis was supported under NASA contract NAS 9-12776

*Independent consultant

TABLE OF CONTENTS

1. INTRODUCTION

1.1 TASK DEFINITION

This project concerns an evaluation of the needs and status of a remote sensing data analysis applications development system (ADS) for the Johnson Space Center's Earth Observations Division (EOD). The state of art in the analysis of remotely sensed data is such that new applications often require new algorithms and techniques, and present methods are not entirely satisfactory. Thus, there is considerable effort being devoted to the development of new techniques and algorithms, both to improve existing ones and to gain a better understanding of the nature of the data.

This task is primarily concerned with the data processing framework wherein such new techniques and algorithms may be economically and efficiently implemented and tested. Specifically, the task definition included the following objectives:

i) Develop a set of design goals for an ADS,

ii) Evaluate ERIPS, ASTEP, LARSYS batch, and LARSYS 3 with respect to these goals,

iii) Develop a recommended approach for a data analysis ADS at JSC.

In this analysis no quantitative consideration has been given to factors circumscribed by the operating constraints of EOD. These factors include:

i) cost of implementing recommended modifications,

ii) system performance and response as a function of number of users,

iii) availability and capacity of hardware,

and iv) specific hardware implementations.

These factors are those which only EOD and other organizational elements at JSC can take into account in establishing an efficient ADS. They have been treated in this report in a qualitative fashion, only noting the relative cost for different solutions. However, such factors were not considered as limiting factors in the development of an ideal ADS. It is important to keep these considerations in mind to form the proper context for the approach to and results of this study.

1.2 REPORT ORGANIZATION

This report consists of six sections: Executive Summary, Introduction, System Analysis, System Evaluations, Recommendations, and Appendices. The System Analysis section details the methodology employed, and it discusses the concepts involved in the hypothetical, ideal ADS. The section on systems evaluations discusses, both in summary and detailed form, the comparisons of each of the four systems (ERIPS, ASTEP, LARSYS batch, and LARSYS 3) with the established design goals and objectives. The recommendations section summarizes the findings of this study and suggests several recommended courses of action for EOD to pursue along with the difficulties associated with each. Finally, the appendices contain (1) a detailed checklist of the design goals and objectives along with the priorities and the ratings of each system for each particular objective; (2) suggested modifications to LARSYS 3 to enhance its utility as an ADS; and (3) definitions of terms used herein.

2. SYSTEM ANALYSIS

2.1 METHODOLOGY

In developing methodology for carrying out the proposed task, it became evident that a direct evaluation of existing EOD software systems, including, for example, feature by feature comparison, would result in a somewhat misleading and possibly biased final evaluation; this is because such an evaluation would only delineate comparisons among existing systems features, and could easily omit consideration of system aspects not implemented or considered in any existing system. Therefore, from the outset, it was decided to take a "top-down" systems approach, wherein an overall set of design goals and objectives were developed independently of any existing system to serve as a framework for the evaluation of each existing system. The design goals represent general areas of interest that any such system should address; however, they are not specific system features, and, as such, they are not prioritizable. Specific system capabilities have been identified as design objectives, which are prioritizable. These design objectives are categorized according to the design goal that they address.

The design objectives were then prioritized according to the needs of EOD. Priority codes of from one to four were assigned with the following meanings:

Priority	Meaning
1	Necessary to achieve EOD program objectives
2	Necessary to achieve a high level of EOD's program objectives

Priority (Cont.)	Meaning (Cont.)
3	Desirable feature
4	Questionable desirability

The various systems were then rated as to how well each system functionally satisfied the requirements of each particular design objective. A rating system was established using the following codes:

Rating	Meaning
5	Exceeds requirements of this objective
4	Meets all requirements of this objective
3	Satisfies most of the requirements of this objective
2	Satisfies some of the requirements of this objective
1	Satisfies only a small portion of the requirements of this objective
0	Does not have any such capability as specified by this objective

These ratings along with the priorities associated with each objective were then used to determine the strengths and weaknesses of each of the systems under consideration. Utilizing this information, it was then determined how the various systems "stood up" against the ideal ADS as specified in the design goals and objectives. It is important to note that this determination was not made by any form of "adding up" the rating codes, but rather was performed by closely examining the strengths and weaknesses of each system as reflected in the ratings and priority codes.

Based on these findings, recommendations were then made for suggested courses of action for EOD to pursue. These proposals are described in detail along with the problems involved in their implementation. In light of the areas not considered in this study, only EOD can make a decision as to which plan is the most practical.

2.2 DESIGN GOALS

Design goals as used herein constitute general areas of interest that a system should address. They are not specific functional capabilities or features of a system, but rather represent general characteristics for which the system should provide support. In general, these design goals pertain to all systems regardless of purpose. The individual nature of a system is reflected in supporting design objectives and a weighting system (priority) indicating their utility for a particular purpose. In this context it is then clear that the design goals are not in themselves prioritizable, but that the weighting of the importance of particular system features is reflected in the priorities assigned to the various design objectives.

Seven design goals were established. They are: (i) combined production and test system in a unified framework. This goal refers to the ability of a system to be used both in a production mode whereby procedures are fixed with the user supplying appropriate inputs and obtaining in an efficient manner the desired output; and a test environment whereby the user may easily, temporarily modify the system in order to develop and test new techniques for use in the system. This feature is particularly desirable for systems employed in areas that are rapidly changing and much effort is being devoted to improving the state of the art. This capability provides the user with all the standard features of the system and allows him to modify only those parts in which he is interested. Thus, much duplication of effort may be avoided.

(ii) Simplification techniques for system maintenance and enhancement. This is merely a recognition of the fact that systems invariably have "bugs", additional or modified capabilities of a system are desired, and new techniques for accomplishing similar objectives are developed. A system should provide, through design and documentation, sufficient facilities to enable both present and future system maintenance personnel to make simply and effectively, the requisite amendments. This implies that comprehensive documentation is an integral part of a system, and that careful attention in the system design phase has been paid to such considerations.

(iii) Data and system management facilities. In any system, measures need to be taken to insure the integrity of the system and at least some of the data sets involved. Such precautions may range from simply maintaining a "backup" to elaborate protocols required to access or modify any such parts. Facilities are needed to guard against unauthorized usage or modification - inadvertent or not - which may cause serious problems with the system or for other users. Additionally, this goal addresses the need for simple and effective procedures and facilities to manipulate data sets as required and to provide relevant information concerning their status and history.

(iv) Graceful degradation capabilities. In designing a system, it is important to give consideration to operating the system in a less-than-ideal or degraded hardware environment. Minor hardware failures - be they parts of main storage, peripherals, terminals, or whatever - should not be allowed to render the system unusable. Provisions should be made in the system for utilizing alternate devices or running in a less than ideal amount of main storage when necessary. The system performance may degrade under such circumstances, but it is important that the system be as operational as possible regardless of the environment. This also addresses the topic of system portability, since, in the event of critical hardware failure or loss, ideally the system should be able to be used on another computer.

(v) Convenience features. A system should be designed with the idea of minimizing the user's efforts in his use of the system. Operational procedures should be as easy and flexible as possible and a variety of aids should be available to assist the user. Detectable input errors should be flagged and the user given an opportunity to recover. Also, the user should not be allowed to get into a situation from which he cannot effectively recover. All output should be displayed in a form suitable for the user's needs, and clearly labelled.

(vi) System measurement and evaluation features. This goal refers to the need by both management and system maintenance personnel of information regarding usage of the system and the associated hardware. Such information should be collected automatically by the system where possible, but user inputs may be necessary in certain areas. Reports on this information can aid in identifying such things as "bottlenecks", over- and under-utilized peripheral devices, the types of processing being performed and their frequency, and even the "quality" of processing. These parameters can enable management and systems personnel to utilize more effectively their resources, to anticipate future needs and problem areas, and to aid in determining the over-all effectiveness of the system.

(vii) Basic systems analysis functions. These are the basic functions that a system needs to perform. They are specific to the system itself. Often, systems are designed with a great deal of emphasis on these functions at the expense of the other design goals. Such systems can cause a myriad of problems and are often unsatisfactory for the task at hand. Admittedly, these functions are required, but care should be taken to insure that the other design goals have been properly addressed in the system.

2.3 DESIGN OBJECTIVES

The design objectives of a system are the specific functional capabilities and supporting features of that system. As such, they constitute a basis upon which a system can be designed. They specify the basic features of a system and, for the most part, are implementation independent. Since they are specific system features, they are prioritizable with respect to the needs of the system developers and users.

In formulating the design objectives of a system, it is important to have a clear, concise definition of the purpose and functions of the system. In the case at hand, an ADS for remote sensing analysis at EOD will be used to develop and thoroughly test new algorithms and procedures for various remote sensing applications. Also, the system will be used to study the nature of the data and the associated problems in interpreting it. The basic desired characteristics of the system include that it be able to efficiently process large amounts of data, easy to use for a wide variety of personnel, reliable, and as flexible and complete a system as possible. In particular, the system must be able to serve two types of users who are designated production and techniques development personnel. The production user (as used here) is the user who wishes to process large amounts of remotely sensed data using existing algorithms and system capabilities. His interest is in applying existing techniques to new applications either to determine the effectiveness of these techniques or perhaps, for relatively small data bases, to process the data as the application requires. Thus, the production user needs a system that allows him to process efficiently large data bases, and to obtain comprehensive results suitable for presentation or further analysis. The techniques development user, on the other hand, needs a system that can be easily modified and allows him to test thoroughly and evaluate new algorithms and techniques. He should be able to add, delete, replace, or modify in an efficient manner any algorithms, while assuring the integrity of the standard system. Facilities should be available to expedite the modification process and to aid the user in testing and evaluating an algorithm's or procedure's effectiveness.

The supporting design objectives for an ADS for EOD are described in appendix A. They are categorized by the major design goal that they support. The priorities assigned to each of these objectives are based on the overall EOD program objectives and are not to indicate any particular order in which the objectives should be implemented in a system. The priority codes and their meanings are given in section 2.1.

2.4 EVALUATION OF THE SYSTEMS AND RECOMMENDATIONS

Each of the four systems involved (ERIPS, ASTEP, LARSYS Batch, and LARSYS 3) were assigned a rating code on each of the design objectives. The rating from one to five indicates how well each system functionally satisfies each design objective. (See section 2.1 for an explanation of the rating scheme). Information sources utilized in rating the various systems include:

i) documentation;

ii) several system presentations and question and answer sessions by NASA and contractor personnel;

iii) live system demonstrations;

and iv) "hands-on" use of some systems.

The ratings assigned to a system based on the design objectives and the associated priorities were then used to determine the strengths and weaknesses of each system. No "totaling" of the ratings were employed to compare the various systems, but rather each system was independently evaluated with respect to the design objectives and priorities, and it was then determined how each system "stood up" as an ADS. (Summary evaluations of each system are contained in section 3.1 and detailed evaluations are in section 3.2).

Utilizing this information, it was then determined which system could best be utilized (with modifcations as required) as an ADS for EOD. Also, major deficiencies of each system that could be overcome were so noted, thus allowing more flexibility for EOD. Finally, several approaches to the development of an efficient ADS and their advisability from the standpoint of this analysis are suggested for consideration by EOD.

3. SYSTEMS EVALUATIONS

3.1 SUMMARY REPORTS

The following sections describe in summary form how each of the systems compared to the ideal ADS. The major strengths and weaknesses of each system as reflected in their ratings based on the design objectives are briefly described. Also, a brief general description of each is included. The detailed evaluations may be found in section 3.2.

3.1.1 ERIPS Evaluation Summary

The Earth Resources Interactive Processing System (ERIPS) was developed by IBM for NASA-JSC. It operates on an IBM 360/75 running under the Real Time Operating System (RTOS), utilizing several very specialized I/O devices. It was designed to enable analysts to efficiently process digitized multispectral scanner data, but it was not intended to be used as an ADS as defined herein. This is exemplified by the fact that most of the coding employed in ERIPS is written in a special purpose assembly language.

ERIPS is mostly suited for production processing, though its processing speed could make it unsuitable for processing large data bases. (It would presently take several hours to classify one ERTS frame on ERIPS using the Gaussian maximum likelihood classifier. Special hardware and some reprogramming could overcome this difficulty.) The interactive imaging capability provides users with an efficient means of viewing and selecting data and results, and the menu formats further ease the user's task. Dynamic report manipulation and reproduction with an on-line hardcopier also facilitates system operation.

Grey level and pseudocolor displays with off-line hardcopying provide users with an effective display and a permanent record of images.

In terms of production processing capabilities, ERIPS is somewhat lacking in the area of data and system management facilities - some data sets may not be shared and insufficient safeguards exist for protecting other data sets. System integrity has proven to be less than desirable in the past, some of which was due to running ERIPS in a "background" environment with other systems. Accessibility has also been a problem owing to this "background" mode and operational difficulties such as tape handling procedures, location of the terminals, and, to some extent, scheduling.

As mentioned earlier, ERIPS runs on an IBM 360/75 under RTOS. (Purportedly, there exists another version running under the more standard operating system OS). This, along with the languages employed (HLAL and PL/1) and the peripherals used, implies that the system is not very portable. Thus a change in the available hardware could severely affect its capabilities. However, much of ERIPS has been written using structured programming techniques, and it is constructed in a modular fashion; both of which would facilitate any reprogramming necessary for hardware changes.

Many of the supervisor routines have been coded to be re-entrant, so that multiple users will share the same supervisor programs; but application routines are not employed this way, and thus each user needs his own copy of these routines resident in storage.

In addition to being tied to the 360/75, much of the coding is specific to some of the I/O devices employed. However, the user is allowed a fair amount of freedom in selecting which of the available terminal devices he will use, so though some devices may not be properly functioning, he can still utilize the system. A batch mode of operation is also available, though it is presently only being used by system maintenance personnel. Thus, a user can utilize the system in a degraded fashion, even if all the terminals are unavailable. Also, a limited checkpoint-restart feature exists, thus facilitating recovery after a system failure.

Documentation provided with the system appears adequate, though the user documentation omits consideration of some troublesome situations. Also, some situations can arise where a user can inadvertently put the system in a non-recoverable state, or where much processing can be lost.

Another deficiency of the system is the lack of adequate systems measurement and evaluation information. Only standard accounting data is collected for users.

In terms of basic systems analysis functions, ERIPS is moderately well-equipped. The image registration is rather crude, linear combinations of channels may not be employed in classification, and the Gaussian maximum likelihood classifier is the only one available. However, the structure of the system is such that many functions could be added (by highly trained system personnel!) with a relatively modest effort.

3.1.2 ASTEP Evaluation Summary

The Algorithm Simulation Test and Evaluation Program (ASTEP) was developed by TRW for JSC. It currently operates on the UNIVAC 1108/1110 under EXEC 8 both in an interactive and a batch mode. It was intended to provide users with a system to conduct experiments and analyze remotely sensed data. Its major drawbacks from the standpoint of an ADS for EOD are inherent fundamental deficiencies associated with efficiently processing large amounts of data and a lack of modification aids to assist users in developing and testing new algorithms and procedures. The strong points of the system include that it is relatively easy to use, the design of the system is relatively simple, and FORTRAN has been employed as the source language.

ASTEP also suffers somewhat from operational difficulties. It requires a relatively large amount of storage, so special permission was required to obtain the needed storage to run on NASA's UNIVAC machines in an interactive mode. Though tapes may be used from the terminal, operational problems make this inadvisable, so the recommended approach is to run a batch job putting the data on a Fastrand drum prior to utilizing the terminal. This is a rather awkward procedure and can cause costly delays.

The lack of an interactive imaging capability and the inability of the system to process large data bases in one pass (it must be done piecemeal) severely hampers the use of ASTEP as a production processor. For techniques development work as envisioned here, its utility is only marginal, despite the program's simplicity, due to a lack of modification aids for the user and insufficient data management facilities. Documentation is not very extensive and the user is "left on his own" as to how to make any desired changes.

The capabilities the system does possess seem well suited for aiding the user in employing the available functions on relatively small data bases, and allowing him to effectively analyze the results and compare the results of different (existing) algorithms. These capabilities include such features as (1) an image difference mapping to aid in comparing results of different classification schemes; (2) the ability to easily generate and utilize arbitrary (Gaussian) class signatures; (3) producing tapes for generating color microfiche images offline; and (4) a variety of other convenience features to ease the user's task in operating the system.

3.1.3 LARSYS Batch Evaluation Summary

A set of programs for remote sensing analysis is available on the UNIVAC 1108 under EXEC 2 at NASA-JSC. Though they are often referred to as the LARSYS batch programs, many of the routines were never (and are not now) a part of LARSYS. The original IBM 360/44 interactive version of LARSYS was converted to run on the 1108 approximately five years ago. Since then, other independently developed routines have been installed as separate programs (e.g., ISOCLS, the table look-up classifier). At present, there is an effort going on to consolidate many of these programs into one system with a standardized input format. However, this report is only concerned with the programs as they presently exist.

These programs, though extensive in number, do not constitute an ADS as we have defined it. Rather, they provide a diverse, relatively undocumented set of programs for personnel to use, with a computer being available for further development or testing. Also, none of the programs have been written with the intention of having arbitrary users modify them. They may not be executed in an interactive environment, and there is no interactive, imaging capability. On the positive side, there are more basic systems analysis functions available here than elsewhere, and most of these programs are written in FORTRAN - a well-known language.

These programs provide few of the capabilities desired for an ADS. There is very little user and programming documentation. Knowledge of how to use or modify some of the programs must be obtained either from other individuals or by "digging into" the programs. The absence of a training program further hampers the utility of these programs.

Since the programs are not part of one system, the input formats differ considerably, adding to the user's confusion. Since many of the programs were coded independently, the style of coding varies widely and no common coding practices were used. Also in LARSYS, portions of obsolete coding still exist in programs, though they are now not used. These points may confuse the user who wishes to modify any of the programs.

Only a batch mode is available for execution of these programs. With no real-time imaging capability, use of these programs on large data sets is limited. In general, the programs are not restartable at arbitrary points which further hampers their utility especially on large data bases. (Intermediate output is available for a number of the processors which serves as input to a run with another processor at a later time.) Only rectangular, aligned fields are allowed, which is often very inconvenient when defining fields.

One of the primary weaknesses of these programs from the point of view of users modifying the routines, is the lack of data set protection facility. One common method of modifying these routines is to update a source tape to create a new source. However, these tapes are not protected and may be inadvertently destroyed. Also, no standard test procedures and cases are available to insure that the programs have been correctly modified. Diagnostic messages are sometimes inadequate, further compounding the user's difficulties.

3.1.4 LARSYS 3 Evaluation Summary

LARSYS 3 is basically an outgrowth of LARSYS 2, both of which were developed at the Laboratory for Applications of Remote Sensing (LARS) at Purdue University. It currently operates under CMS on an IBM 360/67 operating under CP at Purdue. It is a timesharing system designed to enable geographically separated users to both process remotely sensed data and to modify the system to test new algorithms. The system is relatively easy to use with many built-in idiot-proofing facilities and somewhat free-form input. The major shortcoming of LARSYS 3 from the standpoint of an ADS is in the area of modification by arbitrary users of the system. Also, many of the basic system analysis functions are not included.

An extensive hands-on-the-system training program is available. Additionally, lengthy, detailed documentation on the system is readily available. This documentation is well organized and well presented but somewhat lacking in terms of user modifiability of the system.

As mentioned earlier, LARSYS 3 is a timesharing system designed for use at (1) terminals at LARS, and (2) RJE (2780 type) terminals at remote locations. It operates on a 360/67 under CMS under CP in a virtual environment, where each user has his own virtual machine. Three modes of operation from RJE terminals are available: (1) the standard interactive mode, (2) a disconnect mode whereby a user's job continues execution while he relinquishes the terminal, and (3) batch mode where jobs are submitted from the terminal for execution by the batch processor.

An interactive image display screen is available at LARS but not at remote sites. This screen allows for field selection (rectangular, aligned fields only) as well as such display capabilities as unidirectional scrolling and limited zoom-in. Only one channel of input data with less than 768 points/line may be displayed on this screen. Hardcopy output of the image is available.

A variety of user convenience features is included. These include an extensive, informative set of diagnostic messages, a set of utility functions, the ability to reroute or temporarily hold output, and the capability to suspend execution or restart in the classification section. Also, the system handles all of the routine tasks associated with tape and disk usage. An initial input tape to disk load option is not included but would be highly desirable.

A system measurement and evaluation subsystem is used at LARS to report on the utilization of LARSYS 3. Detailed and summary reports are produced. An extensive set of standard test procedures and cases is provided. These are used more to verify correct operation of the programs rather than for debugging algorithms.

As stated previously, LARSYS 3 evolved from LARSYS 2. Essentially, all of the algorithms are the same in both versions. Two classification algorithms are available - a per-field classifier and a Gaussian maximum likelihood point classifier. No image registration or preprocessing functions are included. Only LARSYS 3 format tapes are acceptable as input data.

As a techniques development system to meet NASA-JSC's needs, the major drawbacks of the system are the lack of an on-site interactive image display device and some difficulties associated with temporarily modifying the system. Though the system is somewhat modular in design, some of the routines are excessively long and inherently difficult to modify (obscure though perhaps efficient coding is sometimes employed). A few routines are written in assembly language, but most are in FORTRAN IV. Provisions are made for the user to store some modified routines on his private, virtual disk, but multiple versions of the same routine must be handled in an awkward manner. Thus, a veteran programmer may be required to make some of the modifications a user desires.

3.2 DETAILED EVALUATION REPORTS

This section addresses the detailed findings as to how each of the four systems rated in meeting the established design objectives. Each of the systems evaluations is described with reference to each of the design goals. Not every detail of the ratings is discussed, but rather only the salient features. For the detailed ratings on each design objective please refer to appendix A.

3.2.1 ERIPS Evaluation

3.2.1.1 Combined Production and Test Systems - ERIPS

ERIPS is found to offer some rather outstanding features from the standpoint of production processing requirements. However, in light of techniques development requirements, ERIPS is found to be lacking several key requirements.

ERIPS was initially designed to perform analysis, or production processing on multispectral scanner data. The original design requirements did not include provisions for techniques development support.

The system does employ use of a hierarchial processing structure similar to that described in the design objectives. Such features as the interactive, batch, analytical, and central processing monitors are included in ERIPS. However, rather than having a distinct program separation between the different monitor functions (where each function takes the form of a separate program), all monitor functions are included in the ERIPS supervisor in the form of semi-independent subroutines.

ERIPS employs a highly modular programming structure. Each major processing function takes the form of an executive routine which controls the processing of a series of subroutines. Each subroutine, or module, takes the form of isolated and logically intra-related blocks of programming logic. For purposes of efficiency, reduction of subprograms to small one-function modules has been avoided in certain areas within ERIPS. Input routines have been distinctly separated from algorithms. All input functions are handled by the supervisor. All image output is handled by the individual application programs/subprograms.

ERIPS does not provide a means of maintaining both test (techniques development) and production programs under one single system framework. Therefore, the user is limited to use of production programs which he cannot personally alter from a terminal. ERIPS does not provide an ability for the user to interactively add new programs or subprogram modules to the system.

Accessibility of ERIPS has, in the past, proven to be a significant problem and certain accessibility problems still continue to prevail. First of all, the EOD is not considered a prime user in NASA's Real Time Computation Center (RTCC). Therefore, ERIPS users must continually take a "back seat" to the higher priority mission planning users where scheduling is concerned. At present, EOD must project processing requirements for ERIPS five months in advance, in blocks of time. Each month the EOD must review the allocation of processing time, and each week specific time allocations must be made for each user in order to assure maximum utilization of computer hardware. In many instances, the ERIPS users have not been able to achieve the desired processing results during the time allocated for processing due to the system malfunctions, hardware malfunctions and system crashes caused by co-users. ERIPS software was designed to meet very strict configuration control restraints in order not to interfere with higher priority mission planning applications.

ERIPS does provide access to the system via both image and non-image terminals, and the software and data files are generally accessible without significant difficulty. The ERIPS user is required to go to the tape library prior to a given processing exercise, and select the desired input data tape to be used in processing.

3.2.1.2 Simplification Techniques for System Maintenance and Enhancement - ERIPS

ERIPS offers an impressive array of documentation. The four basic sets of documentation are:

1) Functional Specifications;
2) User's Guide;
3) Program Documentation; and
4) Programmers' Guide

The Functional Specifications offers a notable quantity of useful descriptive information concerning "how the system works" in user terms. The User's Guide describes operational procedures for utilizing the system. Program Documentation deals primarily with a more detailed level of functional capabilities of each program within the system. The Programmers' Guide provides information needed for use by programmers maintaining the ERIPS software.

The ERIPS documentation falls short of design objective requirements in two significant areas. First, the user documentation (Functional Specifications and User's Guide) does not offer a description of the algorithms used in the system. The Program Documentation describes how algorithms are performed, but in programmers' terms - not mathematical terms. The purpose of this design objective is to establish a better user understanding of how processing is executed. Additionally, the user documentation deals somewhat superficially with operational procedures to be employed by users. For certain techniques development staff personnel (particularly new personnel and infrequent users), a more detailed operational description is needed to meet the design objectives.

ERIPS employs a good documentation updating procedure as is evidenced by the fact that the Programmers' Guide has been affected by 15 updates in the past 2 years and the Program Documentation manual has received 3 updates in the past 15 months.

The program and data set naming conventions used by ERIPS do not conform to the guidelines set forth in the design objectives. However, conventions have been established which are generally useful and serve the purposes of ERIPS reasonably well. The main area of disagreement between the existing ERIPS conventions and those set forth in the design objectives is that ERIPS data sets do not include the generating program ID in the data set name which, obviously, would improve audit trails within the system.

Program coding techniques used in ERIPS tend to be quite complex. The program development staff for ERIPS used the High Level Assembler Language (HLAL) to develop the assembly language portion of ERIPS (which comprises ~ 85% of the system). HLAL generates assembler language code of a highly efficient nature based upon parametric input supplied by a programmer. The generated assembly language code, though efficient, is necessarily complex in order to serve the needs of generalized application. All program listings of the HLAL generated assembly language contain extensive comments describing the processing performed. The entire system was developed using structured programming techniques for which HLAL was specifically designed.

In the area of computer hardware and peripheral device independence, ERIPS is found to fall considerably short of meeting the requirements set forth in the design objectives. Since ~ 85% of ERIPS was coded in HLAL parametric statements which resulted in generation of assembler language source code, the system cannot be run on foreign computing equipment without undertaking a substantial conversion effort. Certain key portions of ERIPS incorporate special, highly efficient coding to deal with high-speed input and output on the 2314 disk file. This special coding eliminates the possibility of selecting an alternate I/O device in the event that the 2314 disk drive is unavailable. Other portions pertain to the specialized I/O devices employed.

3.2.1.3 Data and System Management Facilities - ERIPS

ERIPS is found to be substantially lacking several key elements set forth in the design objectives concerning data and system management facilities.

ERIPS does employ for systems personnel a reasonably good control procedure governing updates to the system libraries. However, the established procedure does not include a password protection facility. The user does not have the ability to catalog or delete program modules at all since this is handled only by systems personnel.

No clearly defined operational procedures exist which describe how data sets are created, maintained, or deleted. Of course, data sets are created as they are needed, however, not in a closely controlled environment. Image data sets do not receive sufficient protection. Any user can access, modify or delete any data sets created by any user. Statistics data sets cannot be conveniently accessed by any user other than the user who created the statistics. No provisions have been included in ERIPS to eliminate the possibility of accessing certain confidential image data sets, though RTCC procedures allow for some protection of tapes.

3.2.1.4 Graceful Degradation Capabilities - ERIPS

ERIPS offers several facilities which enable the system to operate in a less than ideal environment. However, some of the points set forth in the design objectives under the graceful degradation design goal are not satisfied by the system. It should be noted, however, that some of these points are not necessarily major considerations from EOD's point of view.

ERIPS currently operates in a Real Time Operating System (RTOS) environment, and an Operating System (OS) version exists elsewhere. Both operating systems offer device independence as a feature of the operating system. However, certain segments of the system use non-standard coding techniques in order to achieve efficient utilization of the 2314 disk file. Therefore, while other on-line devices can be traded off in the event of unavailability of the device, the input and output modules dealing with the 2314 disk file cannot be alternated to other devices without program modification.

ERIPS cannot easily be converted to a computer with less complete hardware and peripherals unless the conversion is to other third, or fourth, generation IBM hardware. The reason for this limitation is again the use of programming languages which will operate only on IBM computers.

ERIPS does offer multiple modes of operation. Basically, ERIPS is an interactive system. Additionally, ERIPS employs a batch mode of operation which is generally used for program testing by the programming staff. The batch facility allows simulation of menu inputs in card form and could be used by techniques development personnel. ERIPS does not offer a disconnect mode of operation.

Most program coding in the ERIPS supervisor is re-entrant or reusable coding. This is particularly advantageous since only one core image of the supervisor need be present to serve any number of terminal users at a given point in time. Of course, this feature reduces the total core requirement to serve multiple users. ERIPS applications programs are not used re-entrantly; therefore, multiple core images of an application are required to serve multiple users for the given application.

ERIPS does not employ the type of terminal handler front-end modules that have been described in the design objectives, though this in itself is not a major factor. Such modules would contribute to the ease of conversion to new hardware or terminals.

3.2.1.5 Convenience Features - ERIPS

ERIPS offers an impressive array of user conveniences, some of which exceed the requirements defined in the design objectives. The ERIPS user has the ability to work with the system in an interactive, timesharing environment. Commands and parametric input to the system are entered by the user via a combination of console keyboard and Grafacon pen entries. The user console consists of two television screens, a keyboard, a Grafacon pen device, a variety of special function console buttons, and a shared color display. One of the two television screens on the user console is an alphanumeric CRT-type device. This screen is used to project menus and to accept user menu notation input. The second screen on the user console is an image screen used for both image display and selection (via Grafacon pen input) of test, training, and miscellaneous fields.

ERIPS does not offer the capability in the interactive mode for the user to input, at the outset of processing, a series of commands and inputs in order to establish in advance static, program path choices. The user must dynamically select each processing operation, wait for that processing to be completed, analyze results, then select the next processing option. In the event of an error (user or program), the user is sometimes unable to "back up" one menu and resume processing.

All menu inputs are edited to some extent for proper content and, in the event of an error, appropriate diagnostic error notation will appear on the menu screen. The user may then re-enter the correct data and continue. ERIPS does not offer the user a computer-generated detailed explanation of the error encountered. Formal documentation of diagnostice messages describing probable causes, remedial action, etc., is in the User's Guide.

Recently, a new checkpoint-restart facility has been added to ERIPS. This facility automatically produces a "snapshot" of system status each time the user passes through certain key menus. In addition, the check-point/restart file can be output to tape from disk for later use. This, in the event of a system crash, allows the user to restart processing at the last checkpoint (or "snapshot"). ERIPS does not offer the ability to use the log file of user commands to bring processing back to the point of the system failure. The user must re-enter all commands and parameters which were entered subsequent to the last checkpoint.

ERIPS offers a load option which assist the user in selection of data, from magnetic tape, to be transferred to highspeed disk storage for processing. This is accomplished by displaying data from the data tape on the image screen and selecting the desired area.

In the area of results interpretation aids, ERIPS excels in such features as the ability for the user to define test and training fields (somewhat arbitrary polygons). In addition to the ability to select both test and training fields, the ERIPS user additionally has the option to select a third field category - miscellaneous fields. In addition, the system has the capability to compute performance statistics. ERIPS offers the user facilities to produce one dimensional histograms of original input data. There is no facility for either two dimensional histograms or for production of histograms on transformations of original input data.

During processing, a variety of reports can be produced at the user's option and upon his request. All reports produced are maintained within the system for the duration of a given processing exercise. In the event that the user may later, during the same processing exercise, need to again study a given report, he may wish to maintain a permanent record of the report, he may select the option to have the report output usinn an on-line harcopy device. This is accomplished by the simple depression of a button on the user console.

Images may be output off-line to microfiche or film in either gray shade (or pseudocolor grey levels). Image output on the on-line hardcopy device is not feasible because the hardcopy device is unable to synthesize a variety of grey levels.

No training program is available with the system, so users must learn "on the job".

3.2.1.6 System Measurement and Evaluation Features - ERIPS

The system measurement and evaluation features described in the design objectives are not satisfied by ERIPS. ERIPS does maintain a semi-detailed log of user input commands and parameters. However, this log is not intended to "feed" a system evaluation program. Rather, its intent is to be used by programming technicians to reconstruct conditions which provoked a program error.

No management reporting facilities of the type set forth in the design objectives are available to the ERIPS users.

3.2.1.7 Basic System Analysis Functions - ERIPS

ERIPS rates moderately high in terms of basic system analysis function. Its major shortcomings here are in the area of preprocessing where only a rather crude image registration capability exists. Also, the system can only select subsets of channels rather than more general linear combinations of channels for use in classification and other applications programs. However, the functions available coupled with the interactive image display screens provide a useful set of functions. In the preprocessing area ERIPS has only an image registration capability. The user selects pairs of points he judges to be the same and then the program computes the mapping function between the two images.

The image manipulation and display capabilities include a variety of features. With the Grafacon pen, the user may select fields in the form of polygons with up to eight vertices. Grey maps may be displayed on image terminals and pseudocolor maps on another eight-color image terminals. Black and white television terminals are used for displaying alphanumeric information and an on-line hardcopier may be used to produce hardcopy reports. No on-line image hardcopy device is available though. Images on the grey level terminals may be scrolled in one direction, and a limited zoom feature is available. Most of the above functions are controlled by a special function keyboard or Grafacon pen entries.

Training of the classifier is accomplished through a clustering routine or user selection of training fields and a Gaussian statistics generator. The resultant statistics (mean vector and standard deviation for each channel) may then be displayed on the image screen.

Feature selection is performed by using the average weighted or minimum divergence criteria. Only a subset of channels may be selected. The exhaustive search procedure is employed in the selection.

Classification is accomplished through the use of a Gaussian maximum likelihood per point classifier. The results of classification may be displayed on the image screens, or a tape may be generated for input to one of EOD's Data Analysis Stations where grey shade and color film can be produced.

3.2.2 ASTEP Evaluation

3.2.2.1 Combined Production and Test Systems - ASTEP

ASTEP was not designed to meet both production and test requirements as set forth in the design objectives. Though certain aspects of the techniques development (test) side of the combined systems concept are well satisfied by ASTEP, few of the requirements of production type processing are provided for. In particular, large data bases must be treated "piecemeal", and the system lacks an interactive imaging capability.

Though ASTEP does employ a hierarchial program and subprogram processing structure, the distinct separation of monitoring function as set forth in the design objectives does not exist within the system. No provisions are included in ASTEP for distinct separation of production and test program modules.

ASTEP is an interactive, timesharing system which utilizes remote teletype style units for user interaction. Though no production processing facilities are included, the system does interact with some users in a test environment moderately well. In the interactive mode, the user employs a dial-up terminal and can utilize either tape or drum input image data. Operational difficulties associated with tape input can, at times, hamper the user's efforts. To utilize drum input, the user must previously have run a job putting the data on the drum, and sufficient drumspace for large image data sets is often not available for storage for any length of time. Thus, accessibility can be a problem. The system additionally employs a batch mode of operation which allows the user to submit simulated menu inputs in card form to govern processing.

ASTEP does not employ any of the features included in the design objectives which deal with the skeleton framework requirements. Therefore, in order for the user to add a new program module to the system, he must:

1) code the program modules;
2) compile the program module;
3) modify the calling program module (from tape or cards) to establish the linkage to the new module; and
4) recompile the calling program module.

It should be noted that items 1 and 2 above would be required under the criteria set forth in the design objectives; however, steps 3 and 4 would be preempted by temporary, or permanent, changes to the System Environmental Control Table/s defined in the design objectives.

3.2.2.2 Simplification Techniques for System Maintenance and Enhancement - ASTEP

For the most part, ASTEP falls short of meeting the design objectives defined for this design goal. While the system does satisfy a high percentage of such requirements as computer hardware independence and documentation updating procedures, other more important considerations related to this design goal are not well satisfied by ASTEP.

Though reasonably good user level documentation is available to the ASTEP user, no formal systems or programming documentation is available. The user level documentation which is available for the system reflects most of the necessary user operational information related to both interactive and batch processing. Both the menus and user variable and parametric inputs are well described. The only important element which is missing in the user's guide is a description of the algorithms used by ASTEP application modules in mathematical terms including a rationale for the approach used. This documentation has been updated in the past to reflect system changes.

ASTEP utilizes the Fastrand drum on the UNIVAC 1108/1110 system in order to provide expedient and efficient processing of images. In order to achieve the efficiency level desired in ASTEP, certain input and output routines use some of the advanced programming features of FORTRAN V. Therefore, ASTEP is unable to totally satisfy the device independence requirements in the design objectives. However, it should be noted that such restraints may very well apply to any system intended to efficiently handle such large data bases as are common to earth resources processing. For the most part, however, ASTEP does employ simplified coding techniques to assist users in understanding the coding and to provide for simple conversion.

In the area of program and data set naming conventions, ASTEP fails to meet the criteria established in the design objectives. No standard is used to control program name assignment. No set procedure exists related to data set naming conventions. Data set names are arbitrarily assigned by the user to suit his own needs at run time.

3.2.2.3 Data and System Management Facilities - ASTEP

As a whole, the design objectives supporting this design goal are not well satisfied by ASTEP.

The design objectives which deal with system integrity are partially satisfied by ASTEP. A procedure exists to control programming changes, correction or enhancement. Generally, control is retained by qualified programming personnel. The password protection, which would prevent unauthorized changes or deletions of program modules, is not employed in ASTEP. Procedures used to modify the system and control permanent program changes are not well-defined nor available in the formal documentation.

Data set integrity is not a feature of the ASTEP system. No password protection as specified in the design objectives has been employed by the system. Any user can conceivably access, modify, and delete any existing data sets. Since data set names are arbitrarily assigned by the user at run time, another user would have to discover a data set name being used by another user before he could tamper with the data set.

ASTEP employs no facilities which provide such capabilities as a user program module listing, or display, nor does the system provide the user with facilities to list all currently maintained data sets. No standardized procedures exist for creating or deleting data sets or for their manipulation. However, some of the basic services and information required may be produced via UNIVAC utility routines. It should be noted that such routines, intended for general use, may provide either more, or less information than is required in a given situation and the method of obtaining it and of presentation may prove less than desirable.

3.2.2.4 Graceful Degradation Capabilities - ASTEP

ASTEP is found to satisfy a notable percentage of the requirements described in the design objectives for this design goal. All ASTEP programs and subprograms are coded in UNIVAC FORTRAN V. FORTRAN V offers some advance programming features not available in ANSI FORTRAN (which is the common FORTRAN language supported by most manufacturers). However, with the exception of certain highly efficient Fastrand drum I/O routines and other minor differences, all coding techniques used in development are basic FORTRAN type instructions. Therefore, conversion to foreign hardware, assuming availability of a FORTRAN compiler, should not prove to be an immense task. The major changes involved would be limited to rewriting the I/O routines which deal with the Fastrand drum, accounting for a change in word length, and modifying some incompatible FORTRAN statements.

ASTEP does offer multiple modes of operation. The primary mode of operation offered the user is the interactive mode. This provides the user with facilities to communicate with the system via a remote, dial-up teletype device. The secondary mode of operation available to the user is the batch mode of operation. This mode allows the user to prepare simulated teletype input data in card form to be submitted to the data processing facility for normal batch processing. No disconnect mode of operation, as specified in the design objectives, is offered by ASTEP.

ASTEP is a fairly modular system which employs program overlays. The overlay structure can be modified somewhat with reasonable ease to operate in a somewhat lesser primary storage environment or in a greater primary storage environment in order to increase processing speeds.

None of the dual-table type front-end terminal handler modules are employed by ASTEP. However, the terminal input and output modules of the system are generally common to most standard terminal devices. Therefore, only minimal or no conversion effort would be required to utilize ASTEP on new terminals of standard variety. The same applies to other peripheral devices employed by ASTEP.

3.2.2.5 Convenience Features - ASTEP

ASTEP, though offering several rather attractive convenience features, fails to offer most of the features set forth in the design objectives.

While ASTEP does not employ the CRT type of user instruction facilities (e.g. light pen-like menu interaction) available elsewhere at JSC, the general menu requirements established in the design objectives are well satisfied by the system. The system employs a technique of prompting the user for desired inputs. User response takes the form of answers to questions output on the user terminal. A reasonable amount of editing is performed by the system to determine the accuracy of user input data. Any erroneous input is brought to the user's attention via terminal output diagnositc messages. All inputs are organized and displayed on the user terminal after final input for a given run and before processing commences, and the user is allowed to correct any erroneous input before processing begins.

ASTEP does offer a "load" option similar to the design objective definition which allows the user to select specific data from magnetic tape or Fastrand drum input and output the selected data onto the Fastrand drum to achieve efficient processing in terms of processing time.

All output reports and images are presented in hardcopy form by ASTEP on a line printer or similar device. No television screening facilities are used by the system. Only rectangular aligned fields may be employed for field selection or display.

ASTEP offers the user the ability to produce an output map which reflects the pixels in which two previously produced maps are found to differ. This feature has been found to be useful as a techniques development aid to determine resultant differences between two algorithms in an expedient manner. This feature could additionally be employed as part of a standard test procedure to validate system upgrades, etc.

The histogram capabilities offered by ASTEP exceed the requirements specified in the design objectives. ASTEP offers the user the option to produce one, two and three dimensional histograms. Three dimensional histograms have generally been found too complex to decipher for practical purposes.

Other useful convenience features offered the ASTEP user include an elapsed CPU time display following completion of each processing option and a news feature which offers terminal display of any news items concerning system changes which are in effect but not yet documented.

ASTEP fails to meet the criteria specified in the design objectives for:

1) simplified checkpoint restart procedure;
2) standard test procedures and cases;
3) utility data input/output packages;
4) auxiliary output device routing feature;
5) dynamic report maintenance and display;
6) training subsystem features.

3.2.2.6 System Measurement and Evaluation Features - ASTEP

ASTEP does not employ any of the system measurement and evaluation features defined in the design objectives, nor does the system employ any similar alternatives.

3.2.2.7 Basic System Analysis Functions - ASTEP

ASTEP is not particularly strong in this area mostly because of a lack of preprocessing routines and an interactive image display device. However, ASTEP does offer a rather extensive set of functions in other areas, including a variety of clustering, classification, and feature selection options.

ASTEP does not have any of the functional capabilities in the preprocessing area such as radiometric or geometric corrections or image registration. ASTEP does not directly possess any of the utility function capabilities listed in the design objectives, though some of these features are indirectly available. At present, four input tape formats are accepted (LARS 1, LARS 2, ERTS, and UNIVERSAL) so the need to reformat is lessened. Tapes may be copied using UNIVAC system facilities, but not in ASTEP. The only way to remove bad data is through the DATDEF option where the user can choose the particular subsets of data with which he wishes to work.

In terms of image manipulation and display capabilities, ASTEP is rather inconvenient. Since no interactive image display exists. the user must employ printed image maps to determine field boundaries (no overprinting capability is available). Also, fields are restricted to being rectangular and aligned with rows and columns of the image. ASTEP does allow the user some choice in how he defines data groups in that he can specify, and pool, areas or, in some cases, he can pick out elements with certain values. Histograms can be generated using one, two, or three dimensions. An image difference map option is available which allows the user to easily compare two images.

Several clustering routines are available in the system. Gaussian statistics and other sometimes useful parameters are available from another section of the program. ASTEP also allows the user to directly enter class signatures, which can be a useful tool for testing algorithms. Displays of these results are somewhat inconvenient since no explanation other than variable names are usually printed with the numbers.

Feature selection can be accomplished in either of two ways: (1) picking a subset of channels from a without replacement procedure that maximizes the average divergence, or (2) generating a linear transformation matrix, B, which maximizes the B-average divergence. The latter generates linear combinations of channels which presently may not be used by the maximum likelihood classifier. Other options in the feature selection section are available to aid the user in evaluating the results.

Besides the clustering routines, two other classification routines are available. One is the standard Gaussian maximum likelihood classifier, and the other classifies by quantizing a single channel of data. Classification maps resulting from these functions may be displayed on the terminal or a tape may be generated for producing film output on the DAS.

3.2.3 LARSYS Batch Evaluation

3.2.3.1 Combined Production and Test System - LARSYS Batch

The set of programs available on the UNIVAC 1108 and 1110 at NASA, JSC, referred to as the LARSYS batch programs, does not well satisfy the objectives associated with a combined production and test system. Since the system is not interactive, it is not very useful for production processing where image display terminals are often needed. Though the system is presently used for techniques development work, its utility here is severely limited. This arises mostly from the many programs that have been written as stand-alone programs and are not part of any larger program. Documentation is very sketchy and input formats vary from program to program. Also, none of these programs were written with the intention of having arbitrary users modify them.

Thus, these programs do not function efficiently as either a production or a test system. For production processing, the user must rely on printed output for image maps to select fields, or he may utilize the off-line color film recorder on the DAS with the output from ISOCLS. He then employs the various programs required for the processing to produce his results (DISPLAY results also may be recorded on film using the DAS). In this process, he is burdened with the chore of submitting several runs and saving the output to be passed later to the next run. Also, he may have to employ several different input formats. So, in all, this is a very cumbersome and error-prone procedure.

For a user to modify or add and test algorithms, he needs to obtain program listings and a copy of the program source tape. After somehow establishing how he can fit his modifications into the program, he can then create a new program tape with which to do his testing. This procedure is basically a multistep, cumbersome one with many built-in delays. The other method for modifying or adding new algorithms consists of requesting LEC programming personnel to do the job. Ideally this is more efficient, but in practice it is very slow. This is mainly due to an insufficient number of programmers available to handle the required tasks. Both of these methods tend to cause the number of programs and versions available to proliferate, thus adding to the user's confusion.

These programs consist of several independent routines of which a version of LARSYS is a part. LARSYS is somewhat modular in design with the different functions being called by a driver. Each of the functions in turn is monitored by a driver routine which calls other routines to perform the indicated tasks. Most of the other programs are similarly structured, though they do not offer the variety of functions available in LARSYS.

In terms of accessibility, the LARSYS batch programs do not rate very high. Since this is a batch system, the user must wait for turnaround which can often be very slow. Terminals may be employed for modifying some of these programs, but severe operational difficulties are encountered when using tapes from these terminals.

3.2.3.2 Simplification Techniques for System Maintenance and Enhancement - LARSYS Batch

The LARSYS batch programs are particularly weak in this area. This arises mainly from having many independent programs not integrated into one system, and the lack of sufficient documentation. Since many of these programs use similar utility routines, when one of these is to be modified, as many as six other routines will also have to be separately modified. This makes modification quite cumbersome.

In terms of documentation, ISOCLS is the only program which provides more information other than user documentation. The LARSYS program itself is not documented to any extent other than periodic modification notices. Other programs such as the table look-up algorithm do have some user documentation. Thus, a new user could be expected to have a difficult time in trying to use these programs.

Most of these programs have been written in FORTRAN V though some utility routines and some table look-up routines are in assembly language. Comments have been employed to a moderate extent to aid in the interpretation of these routines. In the LARSYS program, however, many sections of code (particularly the interactive capabilities) are obsolete: they were left over from the old Purdue version and never removed, though bypassed. This further obfuscates the meaning of the programs.

Though most of the programs have a modular structure to some extent, many of the routines are lengthy and could be broken down into smaller routines for the sake of clarity. Interfaces between programs consist of cards and generated tapes. The interfaces between routines in LARSYS are relatively clean except for the use of common blocks as storage pools. Only mnemonic names are used for programs.

These programs do possess a moderate degree of peripheral device independence. However, use of Fastrand files and assembly language tape I/O routines do somewhat limit this independence. The use of FORTRAN does provide for some computer independence, though special features of FORTRAN V have been employed. The use of assembly language routines does tie the program to certain UNIVAC computers, but these routines are relatively few in number. These computer dependent features are used both for programming convenience and to increase execution speed. Additionally, some of the programs are overlaid so the capability to operate in somewhat less than a desirable amount of storage is available.

3.2.3.3 Data and System Management Facilities - LARSYS batch

The LARSYS batch programs do not offer many capabilities in the area of data and system management facilities. The features that do exist appear to be those available under the operating system rather than specially installed.

The programs reside on tapes, of which there are usually several copies. These tapes are not strictly protected and thus their contents may be destroyed inadvertently. Data sets are usually stored on tape and thus also subject to such deletion. The system does allow for duplication of data sets - in the case of programs, with modifications as desired - but no names other than reel numbers are associated with these data sets. The user is responsible for keeping track of his and other data sets, as no index is available for this purpose.

3.2.3.4 Graceful Degradation Capabilities - LARSYS Batch

The LARSYS batch programs allow some degree of graceful degradation. Except for the use of the Fastrand drum and some assembly language tape I/O routines, most of the coding provides to a large extent for device independence. Both the Fastrand drum and tape are accessed through FORTRAN callable routines which may be modified for other devices if necessary.

Other unique features of FORTRAN V are employed which limit the programs "as is" to UNIVAC computers. Overlay structures are employed to allow some programs to run in a reduced amount of storage. This structure is modifiable so the user (with some difficulty) can expand or, to some extent, contract the storage required. At present, the LARSYS program requires 65 K words of storage to execute.

3.2.3.5 Convenience Features - LARSYS Batch

The LARSYS batch programs are relatively awkward to use despite some built-in facilities to make the user's interaction with the system easier. The difficulty arises mostly from having to use several independent programs each with varying input requirements, and then having to pass results from one program to another. Also, the batch mode of operation is often not a convenient means of obtaining results. The lack of sufficient documentation further hinders operations.

Input procedures in most of these programs have been simplified to the extent that some of the input may be in relatively free format. Some error checking is done at input time to prevent wasting computer time, but, of course, the user must then resubmit the run after correcting the mistake.

Relatively little attention has been paid to diagnostic messages in LARSYS batch. Many of the messages from the old version of LARSYS remain. No particular system for such messages exists, and tracing back to the routine where they occurred can often be very difficult. Since the programs are executed in batch mode, all error messages are fatal. This can result in much wasted man and computer time, especially since no checkpoint-restart capability exists. However, often the user may restart a job using the earlier results and thus not have to rerun the entire job.

The LARSYS batch programs have not been designed for ease in modifiability. No documented standard test procedures other than rerunning past jobs are available to the user for comparing the two runs. Also, no special utility input-output package is available for aiding the user in making his modifications. Other such convenience features that are lacking include a load option, a news feature, and a display of the CPU time at arbitrary points in the program. But here again though, the lack of sufficient documentation is the greatest obstacle for the user to overcome.

The user's task is eased somewhat in the area of results interpretation. Features are provided to aid the user in this task, but some notable deficiencies do exist here. Only rectangular, aligned test and training fields may be defined, and only such areas may be classified. One dimensional histograms may be displayed. Performance statistics include figures relating to performance in test and training fields and a breakdown of how whole areas were classified. Hardcopy output of resulting images are displayed on the printer, or tapes may be generated to produce pseudocolor maps on the DAS film recorder.

3.2.3.6 System Measurement and Evaluation Features - LARSYS Batch

No such features exist in the LARSYS batch programs.

3.2.3.7 Basic System Analysis Functions - LARSYS Batch

The LARSYS batch programs are particularly strong in the area of basic system analysis functions. More functions are available here than in any of the other systems. The disadvantage here, though, is that many of these functions are implemented as separate, stand-alone programs. This means the user must learn the varying input requirements for these programs, and he is left with the task of interfacing the results of one program to be used as input to another. This task is somewhat alleviated since many of the programs merely generate new image tapes and others will output the results in a form compatible for input to other programs. However, the user still has the job of keeping track of "what is what" and where is it.

In the preprocessing area, programs are available for performing radiometric corrections, image registration, and miscellaneous utility functions. The lack of real time user interaction in the first two of these severely limits their effectiveness. No geometric correction or image enhancement programs are presently available.

Several image manipulation and display features are available in these programs but often not in a desirable manner. The lack of an interactive image display device is perhaps the most serious deficiency. Also, the imaging on the DAS must be done off-line from the Univac 1108, though imaging on the SC-4060 microfilm recorder is done off-line at the same facility. All other displays are printer produced. For selecting image subsets, the user is restricted to using pooled, rectangular, aligned fields.

Training of the Gaussian maximum likelihood classifier is aided by the clustering programs ISOCLS which is presently a stand-alone program. A Gaussian statistics calculator is available within the LARSYS program. Resulting displays may be output on the printer and image tapes may be produced for display on the DAS film recorder.

Feature selection in LARSYS batch presently consists of picking subsets of channels by using the average weighted divergence criterion or of picking linear combinations of channels using the feature selection procedure developed at the University of Houston.

Only a per-point Gaussian maximum likelihood classifier is presently available in LARSYS. Subsets of results are displayed on the printer and more useful displays are produced by the function DISPLAY (also in LARSYS). DISPLAY generates classification maps with user defined symbols, applies thresholds, and calculates performance statistics. Both a printer map and a tape for the DAS may be generated.

3.2.4 LARSYS 3 Evaluation

3.2.4.1 Combined Production and Test Systems - LARSYS 3

LARSYS 3 provides, to a fair extent, for both production and test runs in a unified timesharing framework. Production processing from RJE's is hampered by the lack of an interactive image display, which is only available at Purdue. The users' test programs are separated from the production programs, but features for handling and storing these modules are sometimes awkward.

No skeleton modules are available; instead, the user needs to modify the calling routine and perhaps other lower level routines. The modified routines may be stored on the user's P disk (a virtual, private disk) if there is sufficient space and duplicate names do not exist. (In the case of duplicate names, the user must store one of the modules on tape for permanent storage). The installation of these modified routines into the system is automatic for the user: i.e., the system uses all routines on the users' T (a virtual, temporary, private disk) and P disks instead of its own copies. Thus, the user is not bothered with the linkage step. However, the user must be certain that there are no unwanted routines on these disks. At present, there do not exist facilities for the user to store complete, modified, ready-to-execute LARSYS systems, though this ability may be indirectly available.

The LARSYS 3 system is structured around a central processing monitor which activates any of several functional load modules. Each of these modules is a particular user function (e.g. statistics, per point classifier, etc.) These modules in turn have a similar structure using lower level programs and general purpose utliity routines to perform specific tasks. For the most part, card reading and user interaction is separated from analytical processing and performed in special purpose routines. However, most of the output and some of the input processing is performed in routines whose main purpose is processing data.

One of the major strengths of LARSYS 3 is its accessibility. Either on-site standard typewriter terminals or RJE stations with several attached 2741-like terminals may be used to access the system. However, only one image display screen is presently supported. So the user with the need for the interactive display is limited the most in terms of system availability. The need by the remote user for a complete RJE substation may be obviated, but the procedure is rather awkward, (he must create a file to simulate card input and reroute his output to some printer.) In addition, the lack of an initial tape to disk input image load option may hinder utilization by a large number of users due to tape drive availability.

3.2.4.2 Simplification Techniques for System Maintenance and Enhancement - LARSYS 3

LARSYS 3 is an extensively documented system. A systems manual, a 3-volume user's guide, a 2-volume program documentation manual, and a 4-volume test procedures manual are supplied with the system. These volumes are well written and provide a thorough description of the system in most areas. The only area where this documentation is lacking is in terms of user modifiability. Here the user is expected to be familiar with the use of CMS and to have a fairly detailed knowledge of the internals of LARSYS 3 - a desirable goal, but one that will not (probably) be met in practice. The documentation is periodically updated and a news feature is employed for supplying temporary documentation.

Two types of training courses are available: a user's course, including terminal sessions, and a course for systems programmers who will work with the internals of the system. The user's course consists of reading material, a cassette recording, and hands-on use of the system. It is designed to teach the user how to employ the system, but here again techniques for modifying the system are not discussed.

Naming conventions for programs and data sets are not employed for the most part. Mnemonic names are used for programs with SUP appended for supervisor programs, and a few scattered other conventions are employed. Input image data sets are assigned run numbers, but disk data sets are treated in a special fashion. The user has two private (P & T) and two shared read-only (C & S) virtual disks. The private disks provide for data set integrity, but only the P disk may be used for permanent storage. Other permanent storage may be obtained by using tapes.

Most of the routines in LARSYS 3 have been written in FORTRAN, though some are in assembler language. No consistent coding technique such as structured programming has been employed. Though most of the routines are of moderate size - a desirable trait to aid in modification - some of the routines are excessively long and perform several (albeit related) functions. Elsewhere, modular programming has been employed.

Due to the use of FORTRAN, many of the internal interfaces of the programs are rather messy. This is mostly from the lack of a true dynamic dimensioning capability in FORTRAN. In LARSYS 3, one large common block is used by the various functions as a storage pool. Though this serves the same function, it is rather cumbersome and awkward to employ or modify in many situations. Other than this, much care has been devoted in LARSYS 3 to establish clean interfaces between the supervisor and the various functions. Rules have been established and followed on the use of internal common blocks, and modification aids are available to aid the user in changing sets of common blocks.

In the area of hardware independence, LARSYS 3 is both good and bad. It operates in a CP virtual environment under CMS on an IBM 360/67 with some modifications to CMS being required. Thus, it is not transportable to other systems as is. Relatively minor modifications would be required to install the system on an IBM 370 virtual machine (e.g., 158 or 168) operating under VM/70 and CMS, but more extensive modifications would be required to install LARSYS 3 under other more common IBM operating systems (e.g., TSO under VS/2 Release 2) with a virtual capability. To transfer it to another manufacturer's computer would require substantial modifications.

Except for the image display, LARSYS 3 is moderately independent of the perhipherals used. Most of this independence is achieved through the CP-CMS operating system, though LARSYS itself allows the user to substitute some devices himself.

3.2.4.3 Data and System Management Facilities - LARSYS 3

LARSYS offers some nice features in the area of data and system management facilities. The only major weakness in this area is in the inability of the user to utilize disk space (other than his P disk) for permanent storage of various data sets. This forces the user to use tapes for data sets that cannot be stored on his P disk, and thus will require operator intervention and that another tape drive be available. This can be particularly inconvenient when using several versions of the same routine when testing new algorithms.

Data set protection in LARSYS 3 is very good since each user has his own virtual disks that only he and system personnel may alter. If the user desires, he may allow designated other users to read or copy his data sets from his P disk. However, these facilities are not for other data sets, though some similar facilities do exist for image tapes. These include duplicating image tapes and safeguards for "write protecting" these tapes.

An index of all program modules on a user's P disk is available but a complete listing of all modules stored elsewhere is not. Both private and community input data file indices are available in the system. A user may "place" an image tape in either library and assign it a number for future reference. At any time he may list the contents of either library using one command.

Since users of the system may not alter the virtual disk containing the system routines, their integrity is assured. Maintenance of the system is performed only by system personnel.

3.2.4.4. Graceful Degradation Capabilities - LARSYS 3

To a large extent, LARSYS 3 provides for graceful degradation within the framework of operating on an IBM 360/67 under CP and CMS. In many places, the user may employ alternate I/O devices, and he may employ alternate modes of operating the system. These features arise from both the operating system itself and the internal coding of LARSYS 3. The major drawback of the system in this area is that it is tied to a (modified) CMS environment, and thus its use is restricted to a rather small subset of IBM computers.

Within this framework the system offers a large amount of graceful degradation capabilities. LARSYS 3 is not tied to the particular disk or tape drives now in use, or, to some extent, to the terminals being used to access it. However, it is tied to the unique interactive image terminal now in use at LARS. The virtual environment affords some independence of the amount of main storage available so reduced storage may only entail slower execution speed. Also, the overlay structure employed is modifiable to allow for increasing the efficiency of core storage utilization.

The user is allowed a variety of options regarding how he inputs his data and where he receives it. The three modes of operation (interactive, disconnect, and batch) of the system allow the user to utilize the system in a less than ideal terminal environment (e.g. insufficient number of typewriter terminals currently available, malfunctioning line printer, etc.) Also, the user may select other output destinations for his printed output, or even have his output held, and he may either employ a typewriter or the card reader for input, though the former can be rather difficult.

To ease the task of employing different or additional peripherals employed by the user, all such devices should be "front ended"; i.e., they should be interfaced to the system through a module which appears to the calling routine somewhat independent of the particular device in question. LARSYS has a similar feature for terminals in that an IBM 3705 communications controller is used. This allows for a variety of terminals to be accommodated. The interactive image display terminal is handled by a special set of routines, but these include all the logic of the functioning of this device, and thus would be difficult to modify for another device. Neither the line printer nor the hardcopy device are so "front-ended".

Though most of LARSYS 3 is written in FORTRAN IV, it would be a sizeable undertaking to utilize this system on most other computers. Differences in word size, FORTRAN, the assembler language coding used, coupled with the use of CMS facilities create this condition. On another IBM computer of reasonable size, portions of the system could be installed with only a relatively modest amount of effort. However, installing all of LARSYS 3 would only be relatively easy on another 360/67 or an IBM 370 virtual machine (158 or 168) running under VM/70 with CMS available.

3.2.4.5 Convenience Features - LARSYS 3

LARSYS 3 was designed to be employed by a wide variety of users, many of whom would not be very knowledgeable about the internals of the system. Therefore, a variety of convenience features have been included to make the user's interaction with the system as painless as possible. This is particularly true in the area of executing the existing programs, but less so where modification of routines is involved.

User input procedures for executing the program have been made relatively simple but not to the point of employing menus. Instead, keywords and a somewhat free format input have been adopted. These may be input at the typewriter or on cards at the card reader, and the two modes may be intermixed. The reference command is available to explain the meanings of the input commands. The inputs are screened by the system and detectable errors are so noted, and the user is then allowed to reenter the incorrect parameter. Also, the user may elect to enter all his inputs and have the system verify their correctness without actually performing the indicated steps. This allows the user to check out an input deck and make any corrections before actually executing the program.

The user is freed from the task of keeping track of image tapes as the system maintains both public and private libraries and associated indices of all image tapes available. Also, the system performs all the necessary tasks to insure that the proper tapes will be mounted and dismounted when no longer needed. Thus the tape drives are more efficiently utilized, and the user need not concern himself with deciding beforehand what and how many tape drives he will need.

A comprehensive, detailed set of diagnostic messages is an integral part of the system. These are self-explanatory to a large extent and the routine of their origin is indicated in a list of these messages contained in the documentation. There are two types of messages produced: error and information. Informative messages provide the user with information regarding what is presently being done by the system. Error messages are of two types - fatal and recoverable. Most of the recoverable errors are concerned with such errors as misspelled keywords and other input errors. Fatal errors cause the system to abnormally terminate with a storage dump optionally available. (This is not described in the documentation.) How much processing has been lost depends both on the nature of the error and at what point in the processing it occurred. Duplication of errors for debugging purposes must be done manually since no separate input file is maintained for this purpose.

A checkpoint-restart feature does not exist as such in this system. However, the user can utilize the files generated during various phases of the program to recover to some previous point. (This is not always possible.) Also, the user can suspend the classification function, saving the results on tape, and later use this tape to restart. No other functions have this capability though their execution can be stopped.

LARSYS 3 possesses a variety of relatively minor convenience features that ease the user's task in his interaction with the system. Here again, though, emphasis has been placed on features to aid in the execution of the program as is rather than modifying it. Such features include a News option which contains reports from system personnel reflecting system changes and other such information. The elapsed CPU time for a function's execution may be obtained, though not between arbitrary points in the program. Printed output may be obtained on the printer, typewriter, a remote printer, or it may be held indefinitely. Similar considerations apply for punched output. Utility functions are available for duplicating tapes, retrieving data, and printing the ID record on image tapes. One notably absent feature that could be used to speed response time and allow for easier modifications of some sections is a load option to use disk and core storage as intermediate storage of scanner data. LARSYS currently fetches only one line of data at a time from the tape, which is relatively inefficient and a serious constraint for some algorithms. A set of standard test procedures and cases with output is provided to enable the user to verify the correctness of some programs. However, no test data generator for this purpose is available.

Hardcopy results consist of printed output and, at LARS, a camera in front of the image display which accepts both Polariod and 35 mm film. So at places other than LARS, only printer maps (no overprinting facility) of images may be produced as image maps.

Compared to other areas, LARSYS 3 is relatively weak in terms of results interpretation aids. Only rectangular, aligned test and training fields may be employed. Performance statistics can give results on classification of test and training fields, but statistics about other areas are not available. No image difference map generator is available and histograms are only of one dimension.

The training program employed consists of cassettes, manuals, and hands-on use of the system. It serves fairly well as an introduction to the use of the system but does not explain how to modify it.

3.2.4.6 System Measurement and Evaluation Features - LARSYS 3

Though not referred to in the documentation, LARSYS 3 does have a system measurement subsystem. This subsystem collects such information as how much CPU time is being used by each of the available functions, system usage by individual users, and other related information. Reports on this data are then produced for system personnel. Also, a standard accounting system is used to keep track of individual users costs.

3.2.4.7 Basic System Analysis Functions - LARSYS 3

In terms of basic system analysis functions, LARSYS 3 is not particularly strong. Many such functions are not included in the system at all, and of some that are, their capabilities are limited. However, with a few notable exceptions, most of these functions could be easily added to the system without modifying other parts of the system.

In the preprocessing area, LARSYS 3 is severely lacking. No programs exist to perform radiometric or geometric corrections and no image registration feature is available. (Some of these functions are available at LARS but not as a part of the LARSYS 3 system.) Some utility functions are available but all are not part of the system.

The image manipulation and display capabilities of the system differ markedly between LARS and remote sites due to the interactive image display terminal at LARS. The interactive image display is of vital importance to such a system since field selection can be quite difficult using a low quality image display of the sort produced by a line printer. In processing large data bases, this display, if properly implemented, can greatly increase user efficiency and produce more usable results. The display terminal at LARS allows for selecting rectangular, aligned fields only, scrolling in one direction, and a limited zoom-in capability. The display has 16 grey levels with a light pen and function keyboard used for interacting with the display, which holds a maximum of 768 points per scan line. At other terminals, the user must use printer-generated image maps and then supply the start and stop line and column numbers. Histograms and other output are also produced on the printer.

The system is designed around a Gaussian maximum likelihood classifier. Gaussian statistics are calculated and may be displayed. Clustering routines are provided to assist the user in selecting fields and classes to use.

Feature selection is performed using the transformed divergence criterion. Only subsets of channels rather than linear combinations may be selected. The selected channels (no linear combinations of channels allowed) then may be used by either the per field classifier or point by point classifier to classify indicated areas. A special display function then can be used to produce more useful printer image maps and to compute some performance statistics for training and test fields and classes.

4. CONCLUSIONS AND RECOMMENDATIONS

In this analysis, four remote sensing analysis computer systems have benn comparatively evaluated - ERIPS, ASTEP, LARSYS batch, and LARSYS 3 - with respect to the needs of EOD for an applications development system. This was done in a top-down manner in which specific design goals and supporting design objectives for an ideal system were estalished; these objectives were then prioritized according to the stated requirements of EOD program activities. The four systems were compared to the design objectives and ratings were assigned to each system according to how well it satisfied each particular objective. It is important to emphasize that these objectives and priorities were established without any quantitative regard being given to such considerations as:

i) cost of implementing recommended modifications,

ii) system performance and response,

iii) availability and capacity of hardware,

and iv) specific hardware implementations.

Before discussing the results of this evaluation, it is worth restating the context within which this study was made in describing the functions and capabilities of an ideal (for an ADS for EOD) system. Such a system would be used to develop and test new algorithms and procedures for various remote sensing applications. Its basic characteristics should include that it be easy to use for a wide variety of personnel, accessible and responsive to the users, reliable, and as flexible and complete a system as possible. The system must serve two kinds of users: production, and techniques development personnel. The production user needs to be able to efficiently process large amounts of data using state-of-the-art techniques. He requires that results be in form suitable for presentation or further analysis. The techniques development person, on the other hand, needs a system where he can thoroughly test and evaluate new algorithms and techniques. The system thus should be easily modifiable and require the user to only have a minimum of knowledge of the internals of the system. The user should be able to easily add, delete, replace, or modify any of the algorithms in use for his own purposes while assuring the integrity of the standard system.

With this framework in mind, the various systems available were evaluated against agreed design objectives by assigning ratings of 0 (does not have such a capability) to 5 (exceeds requirements for this objective) for each design objective. From these detailed evaluations came a "picture" of how each of the various systems compared as an ADS.

(i) ERIPS possesses several key features, notably an extensive, interactive imaging capability and an abundance of user convenience features. Though ERIPS is highly modular, it was not designed for modification by the user community; most of the coding is in a specially designed assembler language and the programming skill necessary to understand the internals of the system is far beyond the average user.

(ii) ASTEP, on the other hand, was written mostly in FORTRAN V and the coding is relatively easy to decipher. However, no modification aids for the user are available and documentation is not very extensive. Though ASTEP can be run in an interactive mode, the use of tapes is limited by operational difficulties and, thus, system use is limited. Additionally, no interactive imaging capabilities exist, and production type processing is severely limited.

(iii) The LARSYS batch programs were also written mostly in FORTRAN V. However, very little documentation exists on these programs, thus making modification a difficult chore. The most serious problem with these programs is that though many functions are available, they are not in a unified system, which creates a myriad of problems for users and programmers alike. The lack of interactive and interactive imaging capabilities further hampers the utility of these programs.

(iv) LARSYS 3 possesses many of the essential features of the ideal ADS. It too is written mostly in FORTRAN, and extensive documentation is readily available. A variety of modification aids eases somewhat the user's task, but other such features do not currently exist. The system is relatively easy to use, has several modes of operation, and a training program is available. An interactive, imaging device exists at Purdue, but is not supported elsewhere. LARSYS 3 is presently lacking in basic systems analysis functions available, but the structure exists for later adding these.

Thus, in terms of which system comes closest to meeting the requirements for an ADS, LARSYS 3 appears to be the most suitable in principle. If LARSYS 3 is to be used most effectively as an ADS, it is worth examining what modifications are necessary to further enhance its utility, and how difficult would such modifications be to make. This study would indicate that modifying LARSYS 3 in several specific areas would produce an ADS which would satisfy most of the needs of EOD. The major areas of modification include adding more analysis functions, adding a more extensive imaging capability, improving the modifiability characteristics of the system, probably converting the system to run under the IBM Time Sharing Option (TSO), and installing it on an IBM 370/158 or 168 locally. (See appendix B for a detailed list of suggested modifications to LARSYS 3.) These latter two modifications are to allow EOD to have their own system locally with mainline IBM support and file compatibility with other IBM systems (because of using TSO rather than CMS). Compared with operating remotely from Purdue, this would eliminate difficult problems of supporting remote interactive imaging devices, transferring bulk data over long distances, configuration control and future growth of the system, and overloading the system at Purdue. This may well represent the best course of action for EOD in terms of capabilities for satisfying their needs for an ADS.

If the above is not possible, one alternative method would be to provide an interactive image display tied into the LARSYS 3 system at Purdue. This would probably require high bandwidth communication lines and intelligent (perhaps specially designed) terminals to effectively provide this ability over the long distances involved. Other modifications to the system as suggested above could be made to LARSYS to increase its utility. However, difficulties may be encountered in the areas of overloading the system and transportation of data back and forth. Such a configuration would have substantially lower throughput and turnaround capacity, but may be suitable for relatively low volume demand.

Two other possibilities for an interactive ADS suggest themselves: build an entirely new system based on the ideal design goals and objectives contained in this report, or radically modify the internals of ERIPS. Developing a new system based on the established design objectives would be a very costly project both in time and money, but it could provide a very effective means of doing techniques development work. Modifications necessary to effectively utilize ERIPS as an ADS consist of establishing terminals in Building 17 and providing users with the capability to work with the internals of the system. The latter would entail reprogramming all algorithmic routines into a high level language;

providing interfaces to other system routines which would allow users to perform such tasks as menu generation using only the high level language; and adding numerous other capabilities to the system. This approach is not highly recommended since it appears that a relatively large amount of effort must be expended, and the resulting system would still not be entirely satisfactory from the modifiability standpoint.

Modification of either ASTEP or LARSYS batch is not recommended. The basic structures of both of these would not be able to accommodate the necessary modifications. However, parts of these systems, particularly some of the algorithms, could be used with minor modifications in developing a new system or as additional functions in LARSYS 3.

APPENDICES

APPENDIX A

DESIGN OBJECTIVES, PRIORITIES, AND RATINGS

This section enumerates the design objectives, their associated priorities, and the ratings assigned to each system on each design objective. In the case of LARSYS 3, ratings in parentheses refer to how the system rates at Purdue as opposed to utilizing the system from an RJE terminal.

The priorities codes employed are:

1	Necessary to achieve EOD program objectives
2	Necessary to achieve a high level of EOD program objectives
3	Desirable feature
4	Questionable desirability
NP	Not prioritizable

The rating codes employed are:

5	Exceeds requirements of this objective
4	Meets all requirements of this objective
3	Satisfies most of the requirements of this objective
2	Satisfies some of the requirements of this objective
1	Satisfies only a small portion of the requirements of this objective
0	Does not have any such capability as specified by this objective

Priority	Design Objectives	Ratings: ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
	I. Combined Production and Test Systems				
1	A). System should combine both production and test programs in a single unified timesharing framework.	1	2	3	1
1	1). Both production and test programs, or subprograms, should be accessible in an interactive, timesharing environment.	1	2	3	0
1	2). System should provide a means of both processing remotely sensed data as well as modifying existing program modules.	2	2	3	1
1	3). System should provide the user with the ability to select all production modules for processing, or all test modules.	0	0	3	0
2	4). System should provide the user with the ability to combine both production and test programs for certain processing exercises.	0	0	3	0
1	B). System should employ some hierarchical processing structure which employs use of monitor programs, processing programs and subprograms similar to these described in 1)., 2)., 3). and 4). below.	3	2	3	1

Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
		Ratings			
1	1). System should include an interactive processing monitor.	3	1	1	0
1	a). The interactive processing monitor should govern all terminal interface processing.	4	1	1	0
1	a. 1). Menu generation, interrogation and editing should be controlled by the interactive monitor.	4	3	3	0
1	a. 2). Image display and manipulation should be under interactive monitor control.	4	0	0	0
1	a. 3). Microfiche output and line printer output should be under control of the interactive processing monitor.	3	0	0	0
2	a. 4). The ability to modify the System Environmental Control Table/s (SECT) should be under interactive monitor control.	0	0	0	0
1	2). System should employ use of an optional batch processing monitor.	3	3	3	3
1	a). The batch processing monitor should provide a facility for volume program testing and analysis processing.	4	4	4	4

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
NP	b). The batch processing monitor should be used in lieu of the interactive processing monitor for large volume pre-defined processing.				
1	c). The batch processing user should simulate menu type inputs via card media.	4	4	4	4
1	3). System should include an analytical processing monitor.	3	2	3	1
1	a). The analytical processing monitor should govern all basic system functions (see section VII).	3	3	3	1
1	b). The analytical processing monitor should receive processing instructions and data from the central processing monitor.	0	0	0	0
1	4). System should additionally employ use of a central processing monitor which essentially controls all processing.	3	2	3	0
1	a). The Central Processing Monitor (CPM) should perform such functions as:	3	2	2	0
1	a. 1). Passing processing requests from the Interactive Processing Monitor (IPM) to the Analytical Processing Monitor (APM);	4	1	1	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	a. 2). Passing resulting output data from the APM to the IPM;	4	1	1	0
1	a. 3). Accessing all input and maintenance of all output files;	2	3	3	0
2	a. 4). Maintaining all reports produced, in a given processing exercise, so as to provide for dynamic report display during that exercise; and	4	0	0	0
2	a. 5). Controlling all change requests related to the parametric System Environmental Control Table/s (SECT).	0	0	0	0
1	C). In order to provide for testing of unproven algorithms, or techniques, the system should provide a distinct separation of production program modules from test-type program modules.	2	2	3	2
1	1). The APM and IPM should have the capability to call for execution of either production or test program modules dependent upon the type processing being performed.	0	0	3	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
4	2). The APM and IPM may be in the form of two APM monitors, 1 test, 1 production; and two IPM monitors, thereby allowing the CPM to call either the test or production version of either monitor.	0	0	0	0
1	3). An alternative to the duplication of monitors as described in C. 2). above, would be to have a single APM and a single IPM both capable of determing the type of processing being performed and calling either test or production program modules intermixed as required.	0	0	3	0
2	D). Both the IPM and APM should have the ability to call a given number of program modules in excess of those which are active at a given point in time.	0	0	0	0
NP	1). The excess (or non-existent) program modules would best be described as "Skeleton" modules since they provide a "Skeleton" framework in which to add new program modules.				
2	2). The system should provide the ability to compile and catalog a given program module and then activate (or include the new module for processing) by entering a temporary or permanent change to the SECT.	2	0	2	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	3). The system should also provide for de-activation of obsolete program modules by entry of a change to the SECT.	1	1	3	1
2	E). Each major processing function under control of the APM and the IPM should, in effect, take the form of an executive routine which calls sub-program modules in a logical sequence in order to achieve the desired processing.	3	3	3	1
2	1). Each major processing function should have the ability to call all sub-programs, then activated, on the SECT.	4	4	4	4
2	2). The major processing functions should include a subprogram skeleton framework similar to that described for the APM and IPM in D). above, providing for expansion or contraction of subprograms controlled by the SECT.	0	0	0	0
1	F). Accessibility:	1	2	3	1
2	1). System should be available for use without requirement for advance scheduling of computer time for individual users.	1	3	4	2
1	a). Interactive processing should be available on a non-scheduled basis.	2	2	4	0

Priority	Design Objectives	Ratings: ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	b). Batch processing should be available on a non-scheduled basis.	1	3	4	3
2	2). System should employ several types of user terminals for processing.	3	2	2(3)	1
2	a). A non-image terminal of the CRT or teletype style intended for use in techniques development work.	0	4	4	1
1	b). A terminal with image generation capability for use in production processing.	5	0	0(3)	0
1	3). The system software must be maintained in a manner which provides expedient access for processing.	2	3	4	2
1	a). Both source and object programs must be readily accessible to the user during any given processing exercise.	1	3	4	2
1	4). All data files must be maintained so as to provide reasonable access during processing.	2	2	4	3
2	a). Index maintenance of available data sets (described under "V. Convenience Features") will provide reasonable data file access capability.	0	0	4	0

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	b). All data should be maintained in a hierarchial storage structure wherein the most frequently required data is kept in high speed storage (such as drum or disk) and the less frequently required data is kept on a low access speed device (such as magnetic tape).	3	3	2	2
3	5). Data sets should be organized in a fashion which permits simplified access to selected types of data within a given area.	0	0	0	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
	II. Simplification Techniques for System Maintenance and Enhancement				
1	A). System must provide three basic levels of extensive documentation.	3	1	3	1
1	1). Systems documentation must describe the overall systems concepts used.	3	2	3	1
1	a). Description of techniques used to call sub-programs	3	1	4	0
2	b). Description of skeleton program calling techniques	0	0	0	0
2	b. 1). describe System Environmental Control Table/s	0	0	0	0
1	b. 2). describe methods to be used in adding new program modules to the system	2	0	2	0
1	b. 3). describe method to be used to delete a given program, or module, from the system	2	0	2	0
2	b. 4). supply examples of techniques b. 2). & b. 3). above.	2	0	0	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	c). Description of interrelationship between programs and subprograms	3	2	3	1
1	c. 1). complete description of hierarchical relationship between parent and subprograms	3	2	4	0
1	c. 2). description of any pre-requisite processing, by other programs/ subprograms, required for a given program module.	3	2	3	1
1	2). Programming documentation must provide a complete description of all programs and subprograms.	4	2	4	0
1	a). Detailed description of overall functional purpose of each program module	4	3	4	0
1	b). Description of all input and output specifications	4	3	4	0
1	c). Descfiption of all processing performed by each program module	4	3	4	0
1	c. 1). describe all optional processing	4	3	4	0
1	c. 2). describe all algorithms performed	2	3	4	0
1	c. 3). describe any calls for subprograms which may be included in a given module	4	2	4	0

Priority	Design Objectives	Ratings: ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	c. 4). provide complete flowcharts	4	2	3	0
1	c. 5). all program listings must contain extensive comments	3	3	3	3
1	c. 6). detailed description of any prerequisite processing requirements.	2	2	3	0
1	3). User documentation must describe the system, its basic functions and how the system is used in terms suitable for general users.	3	3	4	1
1	a). Each system function must be described in non-computer terms, including a description of the relationship which exists between a given function and other function within the system, and a description of all inputs and outputs.	3	3	4	1
1	a. 1). Any pre-requisite processing required, for a given processing function must be described in detail.	3	3	4	0
1	b). The user's manual must additionally describe, in detail, all techniques available for use in operating the system.	3	3	3	0

Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
		Ratings			
1	b. 1). Each menu, or menu type input facility, must be described including description of all possible (feasible) entries which can be made on the menu.	3	4	4	0
1	b. 2). All error messages which can be encountered must be described in detail reflecting all conceivable causes along with the appropriate corrective action.	3	3	4	0
1	c). Each algorithm used within the system must be explained in detail in purely mathematical terms.	2	2	3	1
1	c. 1). The mathematical description of each algorithm may be accompanied by a program listing of the algorithm. However, the listing must not preempt the preparation of the detailed description of the algorithm in mathematical terms.	2	2	3	0
1	c. 2). All algorithm documentation should be set apart from other user documentation in a separate section of the manual.	1	3	4	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	d). The user's manual should additionally provide training information or pre-requisites.	0	0	3	0
1	d. 1). Manual should either include a programmed instruction type training course, or should refer to a similar separate publication.	0	0	3	0
1	B). Documentation updating procedures must be established in order to assure the current status and accuracy of documentation beyond original system implementation.	3	3	4	0
3	1). Procedures should provide for interim documentation releases via a news option included in the system or other temporary documentation media.	0	3	4	0
2	C). Standardize program and data set naming conventions:	3	1	1	1
3	1). Incorporate program naming convention which	3	1	1	1
3	a). provides ready identification of all programs and subprograms;	3	1	2	2

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
3	b). reflects program type;	4	0	1	0
NP	b. 1). System Monitor Program				
NP	b. 2). Production Program				
NP	b. 3). Test Program				
3	c). defines the sub-system to which the program is related;	4	0	0	0
NP	c. 1). System Monitor				
NP	c. 2). Interactive Sub-System				
NP	c. 3). Analytical Sub-System				
3	d). reflects the program's hierarchical relationship within the system;	2	0	1	0
3	e). includes program number;	0	0	0	0
3	f). includes sub-program number (if a sub-program); and	0	0	0	0
3	g). reflects program/sub-program version number.	0	0	0	0
3	2). Utilize a data set naming convention which	1	0	1	0
2	a). includes complete program name within the data set name;	0	0	0	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	b). additionally includes a unique file, or data set, number, or identifier;	0	0	0	0
1	c). provides a reasonable audit trail reflecting the origin of all data sets.	0	0	0	0
1	D). Simplified Program Coding Techniques:	3	3	3	2
1	1). To the extent feasible, the use of complex coding techniques, for purposes of efficiency, must be avoided so as to provide for ease in understanding by future maintenance technicians.	3	3	3	3
1	a). Simplification of programming techniques must not create processing bottlenecks.	4	4	4	4
1	2). Clean interfaces between program and sub-program modules must be established.	3	3	3	2
1	3). Extensive program modularization:	3	3	3	2
1	a). Program modules must take the form of isolated and logically related blocks of logic in order to simplify the user's task in modifying the system.	3	3	3	3

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	a. 1). Input and output routines must be set up in separate modules, isolated from algorithmic modules.	3	1	2	1
1	b). Program modules should be reduced to small "one-function" sub-programs in preference to larger multifunction modules.	2	1	1	1
1	b. 1). Production modules, due to the volume of processing, should not be reduced to small "one-function" modules, however, production modules must be modularized to the extent outlined in 3. a). above.	3	3	3	3
1	b. 2). Test modules should be reduced to small "one-function" modules in order to allow techniques development personnel the ability to modify a given algorithm sub-function without endangering other unrelated processing.	0	0	0	0
2	c). Each program, or sub-program must have the inherent ability to call a pre-established number of skeleton sub-programs in addition to the active program modules,	0	0	0	0

Priority	Design Objectives	Ratings: ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	d). System should incorporate either the structured programming concept or, perhaps, a standard programming guide-line to assure consistency of program logic without imposing critical bottlenecks.	4	1	1	1
1	e). All programs and sub-programs should contain extensive comments fully explaining the processing being performed.	3	2	3	2
2	E). Parametric activation of new programs modules within the system:	0	0	0	0
2	1). Among other elements, the System Environmental Control Table/s (SECT) must contain an "on-off" type switch which indicates the active/inactive status of a given module.	0	0	0	0
2	2). The user must have the ability to activate, or de-activate a given module via a console keyboard entry.	0	0	2	0
2	a). If such entry is made to change the status of a production module during a test run, the change will be a temporary change for the duration of a single processing cycle.	0	0	4	0

Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
		Ratings			
2	b). Changes to the active/inactive status of a test module may be either temporary or permanent at the programmers' option.	0	0	2	0
1	c). Permanent changes to the status of production modules may only be made when satisfactory password testing has been passed by an appropriately authorized technician.	4	0	4	0
2	d). The SECT will be re-set to its original setting after all processing exercises wherein temporary changes are made.	0	0	3	0
2	e). At users' option, a SECT may be saved on a file to be used with later runs, thereby eliminating need to re-enter redundant input in subsequent processing exercises.	0	0	3	0
1	F). Peripheral Device Independence:	2	3	3(2)	3
1	1). To the extent feasible, programs which use input and output devices must avoid coding which specifies a particular device.	2	3	3(2)	3
1	G). Computer Hardware Independence:	1	3	1	2

Priority	Design Objectives	Ratings			
		ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	1). To the extent feasible, coding techniques which are only available on the original processing computer, must be avoided.	0	3	2	3
1	a). Where an option exists, coding of a more generalized nature, when practical, should be used in preference to coding which would tie the software to a given computer.	1	2	2	2
1	2). Provide capability to execute system in flexible core storage environment.	3	2	3	2

Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
		Ratings			
	III. Data and System Management Facilities				
1	A). System Integrity:	2	2	3	2
1	1). System must employ a standard procedure, a protocol, associated with production program replacement, deletion or inclusion.	3	2	4	1
1	a). No user should be able to catalog, or delete a production module. Program deletions should be performed by maintenance personnel having appropriate password.	3	3	4	2
1	a. 1). System update passwords should be controlled by a senior technician, or group, responsible for accepting or rejecting such requests.	2	2	4	0
1	a. 2). System update passwords should be regularly changed so as to prevent their misuse.	0	0	4	0
2	2). System must incorporate a standard procedure governing the creation, maintenance and deletion of data sets.	2	2	3	1

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	a). Such procedure must assure the integrity of all data sets created, assuring that no data set may be inadvertently deleted by a user other than the user for whom the file is being maintained.	2	1	4	1
3	b). Utilization of a data set naming convention similar to that described in "II Simplification Techniques for System Maintenance and Enhancement" will better enhance the integrity of data sets.	2	1	1	0
2	b. 1). Inclusion of a protocol wherein a given data set may only be deleted concurrent with the "sign-on" of its parent user, or by maintenance technicians submitting appropriate password, would additionally protect data set integrity.	0	0	3	0
1	B). System should provide a technique which allows use of a given data set by users other than the creator of the data set while assuring the integrity of the original data set.	2	2	3	2

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	1). In order to assure the integrity of the data set, when in use by other users, system must "write-protect" the data set to all users other than the parent user.	2	1	3	1
1	2). To provide non-parent users with some flexibility in use of foreign data sets, the system should allow duplication of the data set wherein a new data set name would be used.	2	2	3	2
1	3). System should additionally provide a second data set facility which prevents reading of certain classified data sets.	1	0	1	0
3	C). User program module index display feature:	0	0	3	0
3	1). System should incorporate the ability to display, upon users' request, a complete, paged listing of all program modules currently cataloged under his name.	0	0	3	0
2	D). Index maintenance of available input data:	0	0	3	0
2	1). System should maintain an index, in permanent on-line storage, of all available original input data and interim output data tape files.	0	0	3	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	a). Each output file generated should be cataloged in the index file including tape reel number and date created.	0	0	3	0
2	b). All new input tape files delivered to the computing facility should be entered into the index via keyboard entry.	0	0	4	0
2	2). Upon commencement of a given processing exercise, the user should be able to request a display of all available input files on the terminal.	0	0	4	0
2	a). Upon analyzing the available data file index, the user should also have the ability to request via a keyboard entry that operations mount a given tape file.	0	0	5	0

Priority	Design Objectives	Ratings: ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
	IV. Graceful Degradation Capability				
2	A). Ability to trade-off performance for flexible system facilities:	1	2	3	2
2	1). To the extent feasible, all coding related to reading and writing on I/O devices must provide for device independence.	1	3	3(2)	2
2	a). Allow for use of alternative devices where possible.	2	3	3	2
2	2). To the extent feasible, program coding must use instructions common to equivalent programming languages on other manufacturer's computers.	1	3	2	3
2	a). Use of unique coding capabilities available in only one manufacturer's compiler must be avoided.	1	3	2	3
1	3). Use of computer hardware dependent programming languages must be avoided.	1	3	2	3
2	B). Provide for conversion to computer with less complete hardware and peripherals:	1	2	2	2

Priority	Design Objectives	Ratings: ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	1). To the extent feasible, system must have the capability to process with minimal modification on foreign computer hardware.	0	3	1	3
2	2). System must provide for simplified peripheral device reassignment.	1	2	3	2
2	a). Programs must avoid use of manufacturer dependent I/O processing.	0	2	3	2
1	C). System should employ multiple modes of operation:	3	3	4	1
1	1). Interactive mode of operation	4	4	4	0
NP	a). Provides for user to interact with analytical programs.				
NP	a. 1). User selects processing options one step at a time.				
NP	a. 2). Processing requested is performed and results made available to user.				
NP	a. 3). User interprets results prior to selecting next processing option.				

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	2). Disconnect mode of operation	0	0	4	0
NP	a). Provides user with the same capabilities as the interactive mode with one extension.				
NP	a. 1). User has option to enter a given series of commands then issue a disconnect command, thereby allowing other users to "sign-on" the same terminal while processing is being performed.				
2	3). Batch mode of operation	4	4	5	4
NP	a). Allows user to punch a set of static processing options into cards and deliver deck to computing facility for normal batch processing.				
2	D). System should incorporate program coding techniques which are similar to re-entrant, or re-usable, coding techniques:	2	4	4	0
2	1). System monitor and sub-monitors should utilize such coding techniques in order to provide for minimum core storage utilization for multiple on-line users processing simultaneously.	4	4	4	0

Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
		Ratings			
2	2). Use of such coding techniques must not sacrifice portability to foreign computer hardware.	0	4	1	0
1	E). Incorporate use of program module overlays:	3	3	3	3
2	1). Extensive use of small function program modules will provide for processing in a flexible core storage environment.	3	2	3	3
2	2). Easily modifiable overlay structure should exist to allow user to take full advantage of his system facilities.	3	3	3	3
2	F). Incorporate use of Terminal Handler front-end module:	0	3	3	0
2	1). Terminal Handler must utilize a code conversion table which allows simplified table modification for alternative terminal devices.	0	3	3	0
2	a). Handler should maintain on-line optional Terminal tables to provide for ready switching from one terminal code format to another.	0	3	3	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	b). The Handler must incorporate use of a dual table type code conversion concept wherein both terminal device and computer hardware tables may be changed, or either.	0	3	4	0
NP	2). Terminal Handler will provide simplified changeover to alternative terminal devices when system is transported to a foreign hardware environment.				
NP	3). Employment of a compiler--Compiler may be considered as a possible alternative to the Terminal Handler.				
3	G). System should utilize a special television display handler module capable of simplified modification:	0	0	0(1)	0
3	1). Television Handler module will allow for ease in changing over to new television display devices.	0	0	0(1)	0
3	2). A dual table code conversion concept should be utilized thereby allowing changes to both TV or computer hardware, or either.	0	0	0	0

Priority	Design Objectives	Ratings			
		ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
4	H). System should offer use of a front-end handler to control line printer display format:	0	0	0	0
4	1). Handler will provide simplified technique for change-over to printer with wide, or narrower print line capability.	0	0	0	0
4	2). Should also provide ability to perform code conversions for unique printers.	0	0	0	0
4	I). System should include a special handler capable of performing code conversions for unique hard-copy output devices, plotters and other unique equipment.	0	0	0	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
	V. <u>Convenience Features</u>				
2	A). Menu or other similar interface between user and system:	5	3	3	2
2	1). All user system commands shall take the form of menu type notations.	4	3	2	0
2	2). Menu generator should also have the capability to offer, as an option, the capability to display a brief description of each menu--intended to prompt, or remind, the user rather than for purposes of education.	0	0	3	0
2	3). All user menu responses should be extensively edited for proper data content.	3	3	4	0
2	a). Erroneous input must be so noted on the terminal device allowing the user the opportunity to re-enter his command.	3	3	4	0
2	4). A command language subset should provide user with the option to establish both static and dynamic program paths for processing.	1	0	4	3
2	5). Input procedures should be both flexible and simplified in order to eliminate need for redundant input.	3	3	3	3

Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
		Ratings			
2	6). System should provide ability to "back-up" one menu level, in the event that an error occurs, without loss of results already achieved.	2	2	2	2
1	B). System should produce meaningful diagnostic error messages upon occurence of an error during processing.	3	2	4	1
1	1). All user input commands to the system should be vigorously edited for acceptable data content.	3	3	4	2
1	a). Any input which may result in an abnormal end of job or which may produce undesirable results should be prevented.	3	3	4	2
1	b). The input command editing function should respond with meaningful edit diagnostic messages describing the error encountered along with recommended alternative inputs.	3	2	4	2

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	2). All error messages should be self explanatory and should include a reference number which points to more detailed written documentation describing the error, all conceivable causes along with recommended remedial action.	2	2	4	1
2	a). The documentation on each error message should state which module of the system produced the error message.	0	0	4	0
2	3). System should provide multiple levels of error messages:	2	1	4	1
2	a). Fatal errors; a category of error wherein processing cannot continue.	4	4	4	2
2	a. 1). Storage dump should be optionally available in this event.	2	2	2	0
2	a. 2). Such errors should be clearly described with appropriate user output.	2	2	4	1
2	b). Warning errors; a category of potential error wherein the user is advised of questionable conditions and given the options to either continue or terminate processing.	1	0	3	0

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	b. 1). In the event that the user may elect to terminate processing, a storage dump should be optionally available.	0	0	3	0
2	4). All user inputs should be saved on a separate dump file. This file can be used as the input stream to the program to aid maintenance personnel in duplicating error conditions encountered. When this file is used, a storage dump will be given at the detection point of the error. At the end of a run, the user may delete this file.	2	0	0	0
1	C). Simplified checkpoint-restart feature:	3	1	2	1
1	1). System must maintain a temporary log file of user commands.	3	0	0	0
1	a). In the event of a system crash, after problem is resolved, system must have ability to go into a restart mode of operation.	2	0	0	0
3	a. 1). Menu generator must be able to display the log file of user commands.	0	0	0	0

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
3	a. 2). Menu response interpreter must allow user to select a log sequence number (associated with a given log entry) to re-initialize processing.	0	0	0	0
1	2). In order to provide a checkpoint-restart facility, system should have the ability to produce periodic snapshots of storage, I/O, etc. on an automatic cycle according to elapsed time or volume of processing or at user specified points during processing.	2	0	2	0
1	D). System should offer the ability to save all intermidiate data generated during processing, for both production and test programs, in order to reduce the amount of re-run time necessary to re-create a given condition.	2	2	2	2
1	1). The user should have the optional ability to delete any of his own non-essential data sets via console keyboard, or card, input.	3	3	4	3

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	E). Load option:	4	4	0(1)	0
2	1). System should provide a means by which data may be transferred from large, low speed mass storage (such as magnetic tape) to into main CPU storage or to secondary, high speed, on-line storage (such as disk or drum storage).	4	4	0(1)	0
2	2). System should incorporate an optional initial load operation which provide the user with a means to selectively input data for processing.	4	4	0(1)	0
2	a). Resulting output from the load operation should be placed in a temporary disk file or duration of the processing exercise.	4	4	0(1)	0
1	F). Standard test procedures and cases:	2	1	3	1
1	1). System should maintain a set of standard test procedures intended for use in validation of new production, or test, modules in addition to verification of proper installation in a new computing environment.	1	1	4	1
1	a). System should maintain both the executable test procedure and exemplary output for cross-verification of resultant output.	0	0	3	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	b). Test procedures should be included to test each major processing function in addition to testing the entire production system.	0	0	4	0
1	2). System should provide a means of updating the standard test procedures and uses to provide for natural evolution.	0	0	4	0
1	3). System should provide a test data generator capable of generating a wide variety of test data for use in techniques development.	0	1	0	0
3	G). Utility data input/output package:	1	0	2	1
NP	1). To assist user in accessing data files				
NP	2). To assist user in outputting results				
NP	3). To assist user in manipulating data.				
2	H). Auxiliary output device routing feature:	3	0	4	2
2	1). System should provide user with the, optional ability to designate that output be routed to a wide variety of auxiliary output devices/ terminals.	3	0	4	2

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	I). Hardcopy output feature:	4	3	3	3
1	1). System should provide for output on any black & white or color terminal device to be re-output onto either a black & white hardcopy device or on a black & white/color microfiche printer at the user's option.	3	0	0(2)	0
1	2). System should provide for both alphanumeric and image hardcopy output.	3	2	2(3)	2
2	3). System should provide a "quick-look" hardcopy image output feature which allows the user to produce immediate output.	3	2	2(3)	2
1	4). System should additionally provide a "delayed-look" hardcopy image output feature which can be produced off line, or on line, subsequent to the user's request.	4	2	0(3)	3
3	J). News feature:	0	4	4	0
3	1). System should employ a news feature, as a subset of the menu generator, which can be used to maintain, and display upon request information about any recent system changes which have been installed.	0	4	4	0

Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
		Ratings			
NP	a). Utilization of news feature should be limited to interim documentation to be used only for the length of time necessary for formal documentation to be prepared and distributed relative to a given system update/change.				
3	K). Elapsed CPU Time Display feature:	0	4	3	2
3	1). System should either upon request or automatically display elapsed time upon conclusion of processing of all processing options.	0	4	3	2
1	L). Dynamic analysis and interim result report maintenance and display feature:	4	3	3	3
2	1). System should maintain, throughout a given processing exercise, the entire text of any report/s generated during processing.	4	0	0	0
2	a). Reports should be accessible for display, at user's option, any time later during a "sign-on" period.	4	0	0	0
2	2). System should maintain an index of all such reports which may also be displayed upon user's request.	4	0	0	0

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	3). System may provide, as an alternative to 1). & 2). above, automatic hardcopy output of generated reports.	0	4	4	4
1	M). Result Interpretation Aids:	3	3	3	3
1	1). Test field and training field feature	5	1	3	3
1	a). System should incorporate the ability for for user to define, on a given display screen, training field ploygons to use in conjunction with ground truth data in "training" a given algorithm/s.	4	3	3	3
1	b). System should also provide the user with facilities to define test field polygons in order to test a given algorithm/s on data other than the training field data.	4	0	3	3
1	2). System should employ the capability to compute performance statistics which provide the user with an analysis report reflecting the quality of performance during processing of a given image.	3	1	3	4

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	3). Image difference map generation feature	0	4	0	0
2	a). System should provide a facility to produce an image difference map reflecting only pixels wherein a given image differs from another given image.	0	4	0	0
2	a. 1). Image registration should be accomplished through use of the image registration feature described in section VII A. 3).	0	0	0	0
2	a. 2). Facilities should be included to provide for user specification input describing symbol equivalencies between the two maps being processed.	0	4	0	0
1	4). As an option, the system should provide one and two dimensional histograms of original input data, or any transformations thereof.	3	3	3	3
1	a). System should employ use of a technique whereby one or two dimensional histograms may be generated.	3	3	3	3
1	a. 1). Feature should provide for production of histograms of sections of a given image.	3	3	3	3

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
3	N). System should provide an expansion of the menu generation function which can be used as a training sub-system.	0	0	4	0
3	1). The training sub-system should take the form of a programmed instruction course with provisions to train the student on a step-by-step basis.	0	0	4	0
3	2). System could either display text-type material on the interactive terminal for student review, or employ use of a programmed instruction manual designed to work with the existing system.	0	0	4	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
	VI. <u>System Measurement and Evaluation Features</u>				
3	A). System should produce an extremely detailed performance log which may, periodically, be output on a line printer for management analysis and review.	1	0	3	0
NP	1). The performance log should include such detailed data as sign-on time, sign-off and total elapsed time, user name, programs utilized in processing, number and type of files created, etc.				
3	B). Additionally the system should include the ability to produce summary level reports based upon the log data.	0	0	3	0
3	1). Summary reporting should reflect all processing activities of all system users broken down into separate production and test categorical reports.	0	0	3	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
3	a). Detailed data on the summary reports should reflect such data as period-to date total processing cycles, total elapsed time, average elapsed time, total files created/deleted in addition to core storage utilization data.	0	0	3	0
3	b). Summary level reports should also include year-to-date data similar to the period-to-date information described above.	0	0	3	0
3	c). System should employ the ability to maintain a record of down time, categorically by reason for down time, and include this information on detailed and summary level log analysis reports.	1	0	3	0
3	C). System should provide periodic system dimension and utilization analysis reports in order to better inform management of impending expansion requirements in addition to notification of need to remove obsolete program modules from the system.	0	0	4	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
3	1). System dimension and utilization reports should reflect detailed information related to each program within the system.	0	0	2	0
3	a). Typical detail might include date of program module activation, source language, core storage required to execute, on-line storage required to maintain module, last date executed, utilization frequency, etc.	0	0	2	0
3	b). System should provide information on any secondary storage requirements, for main storage in addition to peripheral storage.	0	0	3	0
3	D). System should provide for maintaining a log reflecting the quality or accuracy of results produced.	0	0	0	0
3	1). Detailed and summary reports including relevant statistical descriptions should be periodically produced.	0	0	0	0

Priority	Design Objectives	Ratings: ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
	VII. <u>Basic System Analysis Functions</u>				
1	A). Preprocessing	2	1	1	3
1	1). Radiometric corrections	0	0	0	2
1	a). Systematically modify data values.	0	0	0	2
1	2). Geometric corrections	0	0	0	0
1	a). Systematically modify the location of pixels.	0	0	0	0
1	3). Image registration	3	0	0	3
1	a). Find a function which maps points in one image of a scene to another image of the same scene.	3	0	0	3
1	b). System should provide the ability to register an image to a map.	1	0	0	3
1	4). Miscellaneous utility functions, examples:	0	2	3	3
NP	a). Reformat data tape				
NP	b). Copy data tape				
NP	c). Remove/replace bad data.				

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	5). Image enhancement feature	0	0	0	0
1	a). System should provide various options for transforming the data to enhance certain feature.	0	0	0	0
1	B). Image manipulation and display feature	4	2	2(3)	2
1	1). Methods for selecting subsets of the data that satisfy certain user-specified criteria, examples:	3	2	2	2
NP	a). Groups of pixels in user specified areas				
NP	b). Groups of pixels having specified data values.				
1	c). Reliable equipment functionally similar to the Graficon pen should be employed to interactively select subsets of a given displayed image.	4	0	4	0
1	2). Various ways to display characteristics of subsets of the data, examples:	4	3	3	3
NP	a). Grey maps of an image				
NP	b). Histograms				
1	3). Pseudo color or grey map of image data:	4	1	0(3)	1
1	a). System should have the facilities to project a color or grey map of input image data on a given terminal device.	4	1	0(3)	1

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	b). System should provide for user interaction with color or black & white image screen via a method similar to that provided by the Graficon pen device.	4	0	0(4)	0
1	c). Non-continguous area pooling feature:	4	3	3	3
1	c. 1). User should have adequate facilities to provide for treating multiple data sub-sets as a single data set.	4	3	3	3
2	4). Image scrolling capability :	2	0	0(2)	0
2	a). System should have the ability, upon user request, to scroll an image on any black & white or color terminal both horizontally and vertically.	2	0	0(2)	0
2	a. 1). Image scrolling should be executed in response to an interactive screen input providing for up, down, left and right scrolling.	2	0	0(2)	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
2	5). Image zoom-in capability:	3	0	0(3)	0
2	a). Facilities should be employed to provide for optional zoom-in capability of a given image on any black & white or color terminal.	4	0	0(3)	0
2	a. 1). Image zooming should be performed in response to plus or minus (magnify or minify) magnification buttons on the user terminal device.	3	0	0(3)	0
2	a. 1. 1). Image magnification should increment or decrement automatically at a pre-established standard rate.	2	0	0(3)	0
2	a. 1. 2). Image magnification numerals $\pm$ should appear, superimposed on the display screen, in conjunction with each image projected.	0	0	0	0
2	a. 1. 3). The system should offer the user the ability to select a point (on the image screen prior to magnification) which shall be the centerpoint of the resulting magnified image.	4	0	0(4)	0

Priority	Design Objectives	Ratings ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	C). Training of the classifier	3	3	3	3
1	1). Clustering facilities should be included in the system to provide for grouping of data.	3	3	3	3
1	2). Calculate necessary statistics of the distinct classes in an image.	3	3	3	3
1	3). Produce a "useful" display of these statistics for the user.	3	3	3	3
1	D). Feature selection	2	3	2	4
1	1). Generate functions that map the data into a lower dimensional space while trying to minimize information loss in the sense of the distinguishability of the classes involved.	2	3	2	4
1	E). Classification	3	3	3	4
1	1). Assigning individual or groups of pixels to a known class or classes using the statistical description of these classes.	3	3	3	4

		Ratings			
Priority	Design Objectives	ERIPS	ASTEP	LARSYS 3	LARSYS BATCH
1	F). Display and analysis of results	3	1	2(3)	3
1	1). For each of the above functions, it is necessary to display the results or information about the results in a compact and informative way so that the user is provided with information that he can use in judging the quality of his results and in deciding what to do next.	3	1	2(3)	3

APPENDIX B

SUGGESTED MODIFICATIONS TO LARSYS 3

1. INTRODUCTION

This section addresses in detail the modifications suggested to further enhance the utility of LARSYS 3 as an ADS. How these modifications should be done is not discussed but it is assumed that they can be implemented in such a way as to make minimal negative impact on other capabilities of the system. Many of the suggested modifications only require, for the most part, additional code, whereas others will entail various amounts of reprogramming. These modifications have varying degrees of utility and it is by no means implied that all are necessary for a viable ADS. Which ones will be made will have to be determined by a careful cost-benefit study utilizing, among other things, the results contained in this report. Basically though, it is felt that the structure of LARSYS 3 is adequate to accommodate the necessary modifications and thus to serve as an effective ADS.

The suggested modifications fall into four categories:

(i) make the system available locally.

(ii) improve the modificability aspects of the system;

(iii) add more basic system analysis functions;

(iv) and add additional convenience features;

Item (i) above is discussed in section 4 of this report. The other suggestions are outlined in the following sections.

2. RECOMMENDED MODIFICATIONS

2.1 Modification Aids

This area is perhaps the most critical area where LARSYS 3 needs to be amended. Presently, temporary modification of the system by the general user is a rather awkward procedure that receives very little attention in the available documentation. Several features are presented that could greatly alleviate this difficulty.

Firstly extensive system level documentation should be added to the currently available documentation. This documentation should contain thorough descriptions of how to modify the system with supporting examples. Such descriptions should also be (perhaps optionally) part of the training course available.

A variety of changes could be made to increase the modifiability of the system. Some of the lengthy routines (e.g.LEARN) should be segmented into several smaller routines, each having one particular function. Computation and I/O functions should be separated and care should be taken to avoid unduly complex coding. Extensive comments should accompany all programs.

More flexible permanent disk storage space should be available to the user to allow him to store (with appropriate safeguards) any modified or additional routines (including those with duplicate names) and to conveniently access and utilize them. The user's own versions of the system also could be stored and "ready-to-run" without any re-linkage. Utilities for assisting the user such as providing an index of programs available and displaying versions currently in use, should be added. A standard set of test data should be readily available along with a test data generator to aid in debugging programs. Other debugging aids such as are available in some FORTRAN compilers should be clearly described and made convenient to utilize. These changes should not severely impact existing programs but will affect the system more on the monitor level.

Other features to allow for easier modification in some cases include an initial image tape to disk load option that brings in an arbitrary subset of the data and the capability to use this data rather than its current one scan-line-at-a-time from the image tape. Not only could this decrease execution time, but it would be more amenable to new algorithms (e.g. those employing spatial information) requiring the data in this form. The addition of "front-ends" for the image display and other output devices likewise could alleviate difficulties encountered with certain algorithms or differing available equipment.

2.2 Basic System Analysis Functions

The structure of LARSYS 3 is such that many basic analysis functions can be readily incorporated into the system. This situation exists because of the rules established and followed in establishing communication between separate functions, insuring relatively clean interfaces and a large degree of modularity, and the extensive documentation available on the system. However, those functions that violate some basic assumptions concerning the nature of the data and how it is to be processed, will be more difficult to incorporate.

In the preprocessing area functions should be added to enable users to effect image registration, radiometric corrections and geometric corrections. Also a variety of utility routines should be available to allow the user to perform such functions as reformatting data tapes or generating test data image tapes. All of these functions should be relatively easy to incorporate into the system and they-especially image registration-would greatly enhance the utility of the system.

Image manipulation and display facilities of the system would be enhanced by the addition of functions to display transformed image data and extensions to the digital display capabilities such as horizontal scrolling and extended zoom-in and zoom-out capabilities. These functions would be quite useful especially in field selection and should not impact other parts of the system.

The addition of a two dimensional histogramming capability both of normal and transformed data, and the ability to specify non-rectangular fields would significantly enhance the utility of the system in training. The histogramming capability should be relatively easy to add to the system, whereas the non-rectangular field capability could entail significant modifications to many sections of the coding.

In feature selection and classification, the ability to select and utilize more general transformations of the data than only subsets of features could provide for more efficient utilization of computer time using more recently developed feature selection algorithms and provide a means for conveniently testing any new such algorithms. This capability would necessitate some modification of existing coding but major modifications should not be required.

A significant improvement to the system's display-of-results capabilities would be the addition of a function to display classification results on a grey level and/or color imaging CRT device, with appropriate hardcopying facilities. Also the addition of an image difference mapping capability and the optional use of overprinting characeters for printer produced images (perhaps standard available packages such as AUTOMAP) would enhance the effectiveness of the system. The latter two functions would require a modest amount of reprogramming in parts of the system but the CRT display capability should mostly be an add-on feature.

2.3 Convenience Features

LARSYS 3 is a relatively convenient system to use. A variety of features are available to simplify procedures necessary to utilize the system. However in certain areas, such features are lacking.

One such area is a restart capability. A more general check-point-restart capability to save necessary disk files and an optional record of all inputs could aid in preventing unnecessary wastes of man and machine time. Additionally the user should be allowed to "backtrack" as much as possible in a processing exercise without losing any necessary results. These features would entail a fair amount of reprogramming but would significantly increase efficient utilization of the system.

Terminal operations could be simplified to some extent in a few areas. These include the addition of more imaging terminals along with extended imaging capabilities. Procedures for employing only the type-writers as terminal input should be simplified to enhance system availability in the event of the card reader being unavailable. Also various tape formats should be acceptable as input without the necessity of converting to a common format before a run is initiated. These modifications would entail various amounts of reprogramming of the system and would moderately increase the efficiency of system utilization.

APPENDIX C

TERMS EMPLOYED

Production system:

A set of computer programs intended for use in large data base analysis processing in order to determine whether the data processing requirements of a given new application , such as a crop acreage survey,can be satisfied with existing capabilities. The set of production programs is maintained in on-line secondary storage under the strict protection of a rigid protocol in order to assure the integrity of the production software.

Test system:

A set of computer programs, similar to the production programs, which are used by techniques development personnel to develop and test new algorithms. These programs are set aside in secondary, on-line storage and classified as test programs in order to allow for modification and test without risking the integrity of the production programs. As referenced herein, a given test program will have a corresponding production counterpart which may or may not be identical.

Applications development system (ADS):

The combination of both production and test systems (as described above) in a unified system framework.

Module:

As used herein, the term refers to a logically related set of computer instruction which takes the form of a subprogram.

Skeleton module:

A non-existent subprogram. A skeleton module may be initialized in such a manner as to cause the calling routine to either call or not call the given skeleton module dependent upon the setting of an internal switch.

Skeleton framework:

A pre-established given set of skeleton modules whose ability to be called (or executed) by a higher level routine is controlled by a set of corresponding internal switches. The skeleton framework generally provides for a given number of modules to be added, during future development or modification projects, requiring minimal recompiling of the higher level routine/s.

System Environmental Control Table/s (SECT):

A table containing, among other conceivable elements, a set of switches which correspond to the then currently established skeleton modules on a one to one basis. These switches are used by the system to determine the active or inactive status of each individual skeleton module as well as to condition the call instructions of higher level routines.

Interactive:

Describes a system wherein the user has the ability to specify input to a given computer exercise on a dynamic basis.

Menu:

As used in this text, the term refers to a method of prompting the user, on a dynamic basis, for the next required imputs along with advising the user of the options from which he may select. A menu may take the form of a full screen display on a CRT device of all options available for a given exercise, allowing the user to "fill in the blanks" in checklist fashion, or the menu may take the form of typing on a remote teletype-type device each individual option, allowing the user to respond as needed.

Performance:

As used herein, the term refers to efficiency of processing in terms of actual processing time.

Graceful degradation:

A term commonly used in the data processing industry to describe the ability of a given software system to continue processing in the event of a failure of a given peripheral device or lack of availability of the desired amount of primary storage for processing. Most systems require a certain amount of primary storage along with secondary storage including, perhaps, a variety of peripheral devices. If graceful degradation features have been built into a given software system, the system then has the ability to process, possibly sacrificing efficiency, in a reduced hardware environment. The term also applies to software failures.