General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Report No. 77-048
Contract No. NAS8-32047

PAYLOAD SOFTWARE TECHNOLOGY

SOFTWARE TECHNOLOGY DEVELOPMENT PLAN

(NASA-CR-150143) PAYLOAD SOFTWARE TECHNOLOGY: SOFTWARE TECHNOLOGY DEVELOPMENT PLAN (M&S Computing, Inc., Huntsville, Ala.) 133 p HC A07/MF A01 CSCL 09B

N77-27750

Unclas
G3/61 36694

June 22, 1977

Prepared for:
George C. Marshall Space Flight Center
National Aeronautics Space Administration
Marshall Space Flight Center, Alabama 35812

M&S COMPUTING, INC.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

1. INTRODUCTION

The Software Technology Development Plan is the prime end-item of the basic Payload Software Technology Study (NASA Contract NAS8-32047). This plan was prepared by M&S Computing, Inc., for Mr. John Capps, COR, EF-15, Marshall Space Flight Center. As shown in Figure 1-1, the Software Technology Development Plan was derived from a study composed of three consecutive phases. The purpose of the first phase was to identify software technology drivers. The second phase consisted of a detailed analysis of the technology drivers, from which came a list of potential solutions - Technology Items. Phase III provided a critical assessment of the technology items, in terms of development cost, time and priority. The most promising (or urgent) items were subsequently expanded into individual technology development plans for incorporation into this report.

1.1 Purpose

The Software Technology Development Plan was formulated to define programmatic requirements for the advancement of Software Technology. The initial study was designed to identify drivers and suggest solutions to meet the requirements of space flight in the 1980-1990 time period, and to project potential technological needs from 1990 through the end of the century.

All identified software technology requirements were reviewed to establish the following: which requirements can be fulfilled within current state-of-the-art technology; which technology requirements can be developed with a high probability of success and a significant payoff in benefits; and which requirements pose risks (in cost and success) but are considered mandatory for other reasons. The resulting software technology development items are compiled into this development plan. The plan describes justification, cost and schedule, potential payoff benefits, measurable milestones, methods of approach, and methods for verification of results.

A large number of Software Technology Items were identified; however, it does not make sense for NASA to duplicate technology development which is being satisfactorily performed by industry or other government agencies. Therefore, this plan is dedicated to fresh ideas or known items which are not being worked on. Goals have been established by the Outlook For Space (OFS) and the Office of Aeronautics and Space Technology (OAST) Workshops. This plan can only identify the next step toward these goals. Hopefully, it will be a major step.

1.2 Scope

The primary emphasis in the Software Technology Development Plan is onboard processing in a real-time environment. Payload analysis was based on projected Space Transportation System (STS) payloads as described

PAYLOAD SOFTWARE TECHNOLOGY
STUDY PLAN

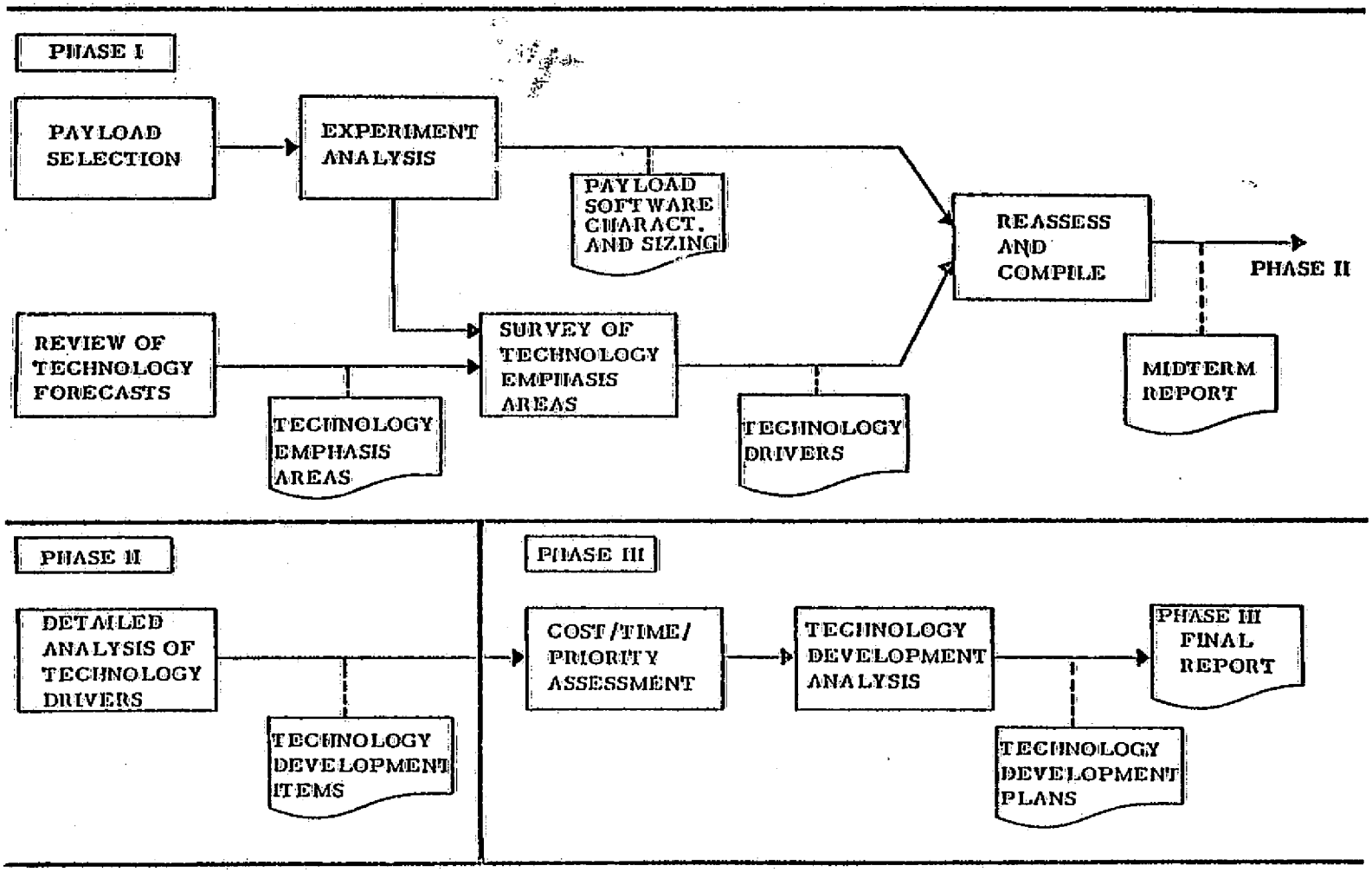


Figure 1-1

in the July 1975, Shuttle Sortie Payload Description (SSPDA), and from OFS and OAST space technology forecasts. This data was supplemented with an in-depth review of detailed payload studies, Design Reference Mission Analysis (DRM), Integrated Mission Analysis and Planning (IMAP), user's information, previous space programs and studies, and in-house expertise.

The technology development plans resulting from this study must be implementable during the 1980's in order to meet the requirements in the 1990's. It is assumed, for this study, that the primary development vehicle during this time will be the Space Transportation System. Therefore, payload analyses performed during this study were limited to those payloads currently planned to be carried on the Space Transportation System.

Finally, it should be noted that this study was primarily concerned with software technology; not with computer systems technology. A specific effort has been made to limit the consideration of Technology Items to pure (or near-pure) software technology development. No attempt was made to analyze future Data Management Systems, as a whole, to drive out technology requirements. Such an effort certainly has merit, and might result in software technology requirements; however, many software technology requirements can be identified without an end-to-end data management analysis.

1.3 Organization

This plan is organized into three sections with three accompanying appendices as summarized in Table 1-1. Section I is standard introductory material and is self-explanatory.

Section II is an executive summary providing an overview of the remainder of the document. It gives a brief summary of the Software Technology Development studies contained in the plan, provides an overall schedule, and identifies any critical sequential or complementary relationships. This section briefly summarizes related technology goals, either in hardware or software items, being pursued elsewhere. It also lists the priority relationships existing so that the plan can easily be adjusted to varying budgetary conditions.

Section III gives a detailed description of each selected Software Technology Development study. Each Software Technology Development Item is allocated a subsection. Each subsection describes the causes leading to the technology need, and relates the problem to the supporting data accumulated during Phases I and II of this study. It defines the magnitude of potential cost or performance benefits, as well as a summary of the anticipated costs of associated technology development.

This section also describes the apparent technological solutions and associated technology development requirement(s). Associated hardware or systems technology items are described to ensure that all facets of the problem, as well as its solution, are clarified. Next, the potential approach that is likely to accomplish the stated technological objective is outlined. Subsequently, schedules and resources required to accomplish this particular technological development item are identified. Available means of intermediate and final measurable verification of the results associated with this item are listed.

The appendices provide a description of the technology requirements from which the Technology Development Studies were selected. The technology requirements are also related to the experiments from which they were derived wherever relevant.

PHASE III REPORT OUTLINE

SOFTWARE TECHNOLOGY DEVELOPMENT PLAN

SECTION I - INTRODUCTION - This section describes the purpose of this plan, the scope of the plan, and the manner in which the plan is organized.

SECTION II - SUMMARY - This section is an executive summary of the Software Technology Development Plan and its derivation.

SECTION III - DESCRIPTION OF SOFTWARE TECHNOLOGY ITEMS - This section is the description and planned approach to each of the Software Technology Development studies selected.

3.1 Item TI-01

.
.
.

3.n Item TI-n

APPENDIX A - Technology Driver Derivation

APPENDIX B - Software Technology Item Derivation Worksheets

APPENDIX C - Cost/Time/Priority Assessment

Table 1-1

1.4 References

A thorough survey was made of current literature as well as government, industry, and other publications relating to software technology. The prime forecasting references used in this study are as follows:

- Outlook for Space, NASA Document, January 1976.
- A Forecast of Space Technology 1980-2000, January 1976.
- OAST Space Theme Workshop, NASA Document, April 1976 (Quick-Look Comments and Working Papers).
- Space Electronics Technology Summary, March 1976.
- OAST Summer Workshop, NASA Document, August 1975.
- Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's (CCIP's), 1975.
- Findings and Recommendations of the Joint Logistics Commanders Software Reliability Work Group, November 1975.
- "Key Developments in Computer Technology: A Survey," IEEE Computer Magazine, November 1976.
- IEEE Proceedings, Second International Conference on Software Engineering, IEEE Catalog No. 76CH1125-4C, October 1976.
- "New Directions in Space Electronics," Peter R. Kurzhals, Astronautics & Aeronautics Magazine, February 1977.
- Software Technology Present and Future, Bobby Hodges, 14th Annual Southeast Regional ACM Conference, April 1976.

2. SUMMARY

This section provides a top-level view of the Software Technology Items identified in the study, a composite development schedule, relationship to other technology developments and an identification of development priorities. The Software Technology Items described in this section were selected in order to support efforts to:

- Reduce the high cost of meeting flight software reliability requirements.
- Remove limitations on data compression and information extraction techniques.
- Increase the capacity of space-to-earth communication links.
- Meet the requirements for human control of complex instruments and remote systems.
- Simplify, reduce the cost of and make usable complex software systems.

2.1 Description of Software Technology Development Items

Phase I of this study concentrated on evaluation of the processing requirements (sizing, speed, input/output rates, etc.) of representative space missions, primarily Spacelab, anticipated for the coming decade. This part of the study provided the basis for selecting drivers of software technology. Consistent with expectations, problems were not peculiar to specific missions, instruments or applications; they were common to many classes of missions, instruments, and applications. Software technology advancements do not just require a new algorithm or unique approach to a given problem. Software technology advancement needs tend to fall in broad, general improvement categories as follows:

- Methods for containing, controlling, and reducing data rate and volume.
- Methods of improving information content.
- Techniques to reduce cost of software development.
- Techniques for improving software reliability and simplifying verification/validation activity.

- Techniques to capitablize on the potential offered by breakthroughs in hardware technology.

- Knowledge of how to use the new hardware technology.

The problems and needs thus identified were thoroughly assessed and compiled into a list of fifteen Software Technology Drivers as shown in Table 2-1 and described in Appendix A. The Technology Drivers associated with Software Development must be considered the most important. Unless significant improvements are made in this area, all other related technologies will be stunted because the associated software will be too costly and/or too unreliable. In Software Systems Architecture, most of the Technology Drivers are related to LSI technology; i.e., the availability of very low-cost processing systems. These systems are particularly suitable for onboard usage and therefore receive significant emphasis. The remaining Technology Drivers are primarily based on anticipated increases in onboard processing of image type data, as well as on anticipated increases in the rates of data to be acquired.

In Phase II of the study, the fifteen Software Technology Drivers were analyzed in detail in search of potential solutions to the problems posed. This analysis resulted in identification of over sixty Software Technology Items. Twenty of these were determined to require forced advancement by NASA if current NASA plans are to be met. The twenty Software Technology Items are shown in Figure 2-1 as related to the Technology Drivers from which they were derived. (Blocks containing asterisks represent relationships established during Technology Development Analysis activities of Phase III. Asterisks in the TI-19 row indicate secondary benefits that may be derived from use of software prototyping.)

The total number of Technology Items identified may be impossible to implement within the resource constraints. Therefore, during Phase III, cost estimates and priorities were assigned to the twenty Technology Items. The result is a recommended set of Software Technology Development Plans. These were obtained through evaluation and selection of the most cost effective set of Technology Items. The interrelationship of the Technology Items and dependencies on other technologies are as described in the remainder of Section 2.

TECHNOLOGY DRIVER SUMMARY

- SOFTWARE DEVELOPMENT
  - SOFTWARE DESIGN ENGINEERING
  - TREND TOWARD S/W DEVELOPMENT BY NON-PROGRAMMERS
  - FAULT-FREE SOFTWARE
  - APPLICATION-ORIENTED LANGUAGE DESIGN METHODOLOGY
  - LOST COST DEVELOPMENT OF AOL COMPILERS
- SOFTWARE SYSTEMS ARCHITECTURE
  - (DISTRIBUTED) SYSTEM PARTITIONING/ INTERCONNECTION TECHNIQUES
  - VERY LARGE STORAGE ACCESS SIMPLIFICATION
  - SOFTWARE FAULT (OWN OR INDUCED) DETECTION
  - SOFTWARE RECOVERY (AFTER FAULT DETECTION)
  - HIGH-SPEED BUFFERING TECHNIQUES
  - DESIGN AND CONTROL OF ADAPTIVE SOFTWARE PROCEDURES
  - USE OF "NATURAL" COMMUNICATION METHODS
  - EFFICIENT LARGE ARRAY SEARCH AND SORT PROCEDURES
  - PARALLEL PROCESSING TECHNIQUES
  - EFFICIENT LARGE ARRAY MANIPULATION PROCEDURES

Table 2-1

TECHNOLOGY DRIVER-ITEM CORRELATION TABLE

TI NO.	SOFTWARE TECHNOLOGY ITEM	SOFTWARE TECHNOLOGY DRIVER: PS-														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	REQUIREMENTS DECOMPOSITION AND STRUCTURING GUIDELINES														*	
2	REQUIREMENTS-TO-CODE TRANSLATION AIDS															
3	SIMPLIFIED SOFTWARE DEVELOPMENT SYSTEM DEMONSTRATION MODEL															
4	QUERY-GUIDED IMPLEMENTATION METHODS															
5	STANDARDIZATION CRITERIA FOR NASA SOFTWARE															
6	AOL DESIGN GUIDELINES/STANDARDS															
7	AOL COMPILER GENERATOR COST FACTORS															
8	INTERFACE CRITERIA FOR NASA DISTRIBUTED PROCESSOR APPLICATIONS	*														
9	TASK CONTROL STRUCTURES FOR DISTRIBUTED PROCESSOR ENVIRONMENTS															
10	PROGRAM ORGANIZATION METHODS FOR REAL-TIME FAULT RECOVERY						*								*	
11	ADAPTIVE HIGH SPEED BUFFERING TECHNIQUES FOR DYNAMIC NETWORKS											*				
12	CONTROL STRUCTURES FOR ADAPTIVE SYSTEMS															
13	IMPACTS OF NATURAL COMMUNICATION METHODS ON NASA PAYLOAD SYSTEMS															
14	ADAPTIVE SEARCH AND SORT PROCEDURES											*				
15	RESTRUCTURED IMAGE ANALYSIS SOFTWARE FOR PARALLEL PROCESSING															
16	OPTIMAL LARGE ARRAY PARTITIONING PROCEDURES															
17	CLASSIFICATION OF ERROR MECHANISMS															
18	GUIDELINES FOR DESIGN DOCUMENTATION CONSISTENCY															
19	SOFTWARE PROTOTYPING METHODS		*	*			*	*	*	*	*	*		*	*	*
20	SOFTWARE TECHNOLOGY MONITORING															

Figure 2-1

Analysis of the Software Technology Items resulted in the selection of seven which represent the highest priority and are most apt to result in the earliest return of appreciable benefits. These are:

- TI-01 Requirements Decomposition and Structuring Guidelines.
- TI-08 Interface Criteria for NASA Distributed Processor Applications.
- TI-09 Task Control Structures for Distributed Processor Environments.
- TI-10 Program Organization Methods for Real-Time Fault Recovery.
- TI-15 Restructured Image Analysis Software for Parallel Processing.
- TI-17 Classification of Error Mechanisms.
- TI-20 Software Technology Monitoring.

The first five items are very specific and are directed to the major goals of reducing cost, enhancing or enabling functions, improving reliability and reducing the raw data flow. The last two items are more general in nature. TI-20, Software Technology Monitoring can and most likely will result in significant contributions to all major goals. TI-17 could very well be considered a subset of TI-20.

2.2 Relation to Other Technology Development

Dependencies exist between variable independent Technology Items. Solution of a problem posed by a Technology Driver requires the full consideration of the interrelationships between these variables as well as the direct relationship to a specific item. This interrelationship is shown in the Item-to-Item correlation matrix of Figure 2-2. In this chart relationships are shown which do not necessarily appear in the Driver-Item matrix, because secondary relationships are identified wherever possible. The primary goal of this chart is to aid in establishing the overall technology Development Schedule described in Section 2.4.

Other software technology development activities are either underway or have been proposed, both within and outside NASA. The more significant of these are discussed in the Software Technology Item Derivation Worksheets (Appendix B).

SOFTWARE TECHNOLOGY
ITEM-TO-ITEM CORRELATION

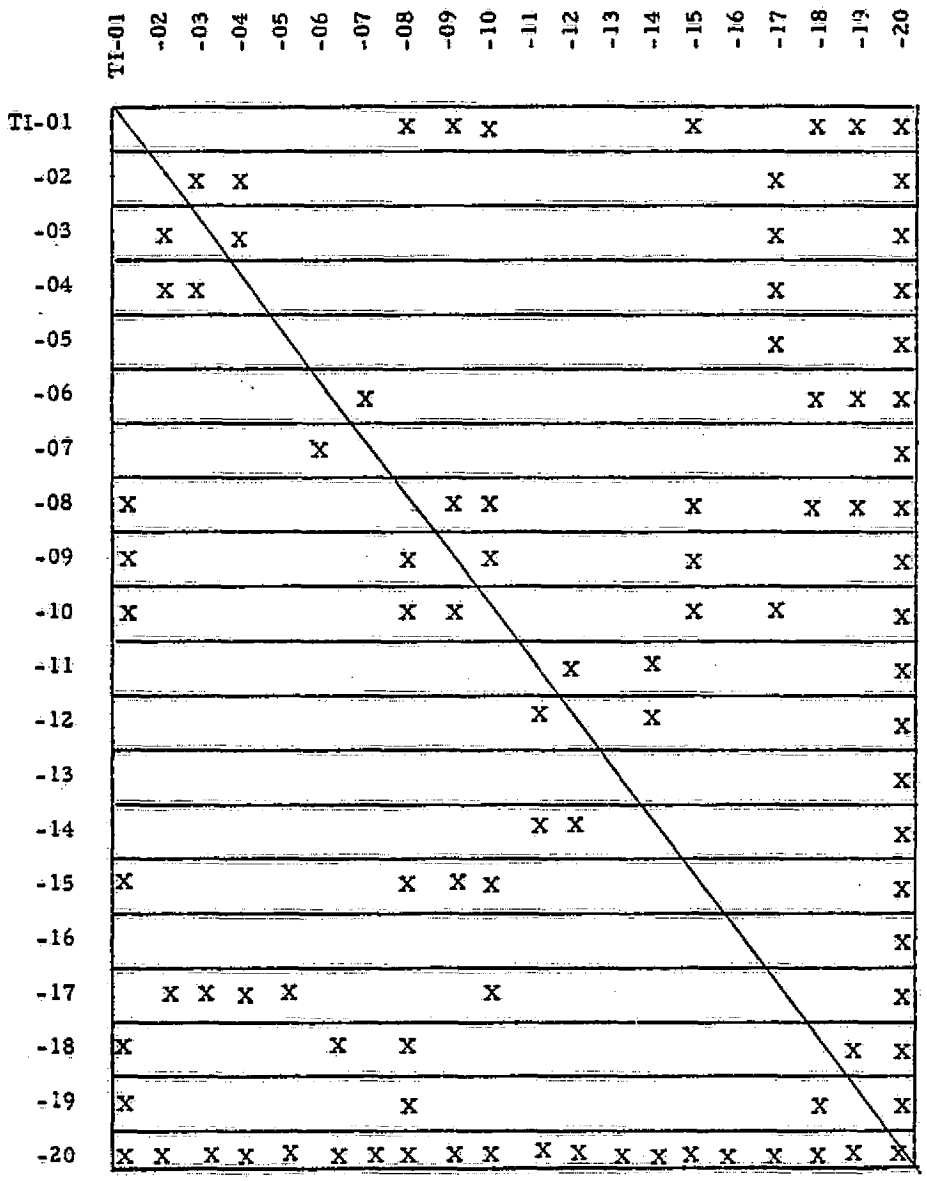


Figure 2-2

2.3 Identification of Priorities

Many factors were involved in setting the priorities of the candidate Software Technology Items. Once the initial step of identifying a problem was completed, the next step was to evaluate the current and estimated state of the art in the associated technology area. This evaluation is reflected in Appendix C, the Cost/Time/Priority Assessment. The levels established were based on the following criteria:

1 - Basic principles observed and reported.

2 - Conceptual design formulated.

3 - Conceptual design tested analytically or experimentally.

4 - Critical function/characteristic demonstrated.

5 - Component/section tested/reviewed in relevant environment.

6 - Prototype/draft tested/reviewed in relevant environment.

7 - Final product tested in operational/space environment.

8 - New capability derived from a much lesser operational model.

9 - Reliability upgrading of an operational model.

These state-of-the-art levels were not used for setting priorities per se, but were used as aids to help determine whether appropriate advancement could be achieved by industry alone, or if NASA participation would be required.

In order to facilitate setting priorities, attain consistency, and limit subjectivity, a logic flow was developed as shown in Figure 2-3. This flow embodied the major elements of priority assignment and resulted in a rating of 0-6 for each technology item. The elements used were:

- Enablement.
- Cost Reduction.
- Enhancement.

Enablement was subdivided into two categories, critical and desirable. (Initially, enablement was divided into mandatory and desirable, but as it became evident that none of the items would actually preclude a space mission, the mandatory rating was downgraded to critical).

TECHNOLOGY ITEM PRIORITY

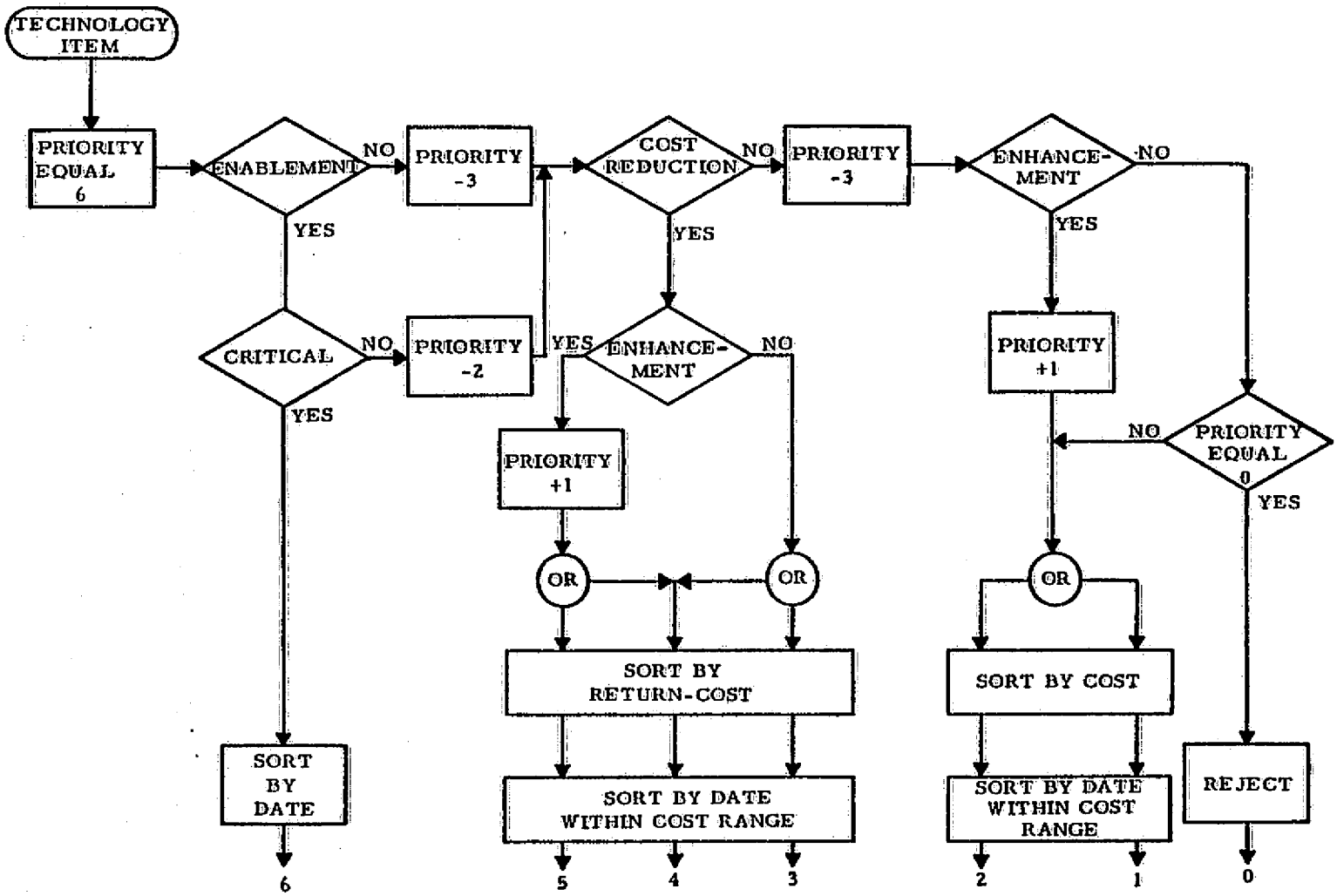


Figure 2-3

Critical enablement describes an item which could gravely affect the achievement of a major NASA objective (e.g., 1000x data at 1/10th cost) if it were not resolved. Critical enablement was given a weight of 6. Desirable enablement describes a technology that would enable a new function to be performed, or allow additional experimentation with a given instrument, that would significantly increase the return benefit of a payload. Desirable enablement was equated in value to enhancement, and thus given a weight of 1.

Cost reduction applies to those software items which could have major effect on the cost of space activity in general. Most of these items fall in the category of Software Engineering or Software Development. Cost reduction was given a weight of 3.

Enhancement applies to those items that would enhance the total information gathering and distribution process. This includes improved reliability, more effective processing, higher quality data recovery, and other advancements that would be highly beneficial. Enhancement was given a weight of 1.

After the initial assessment, other factors (Drivers affected and Cost/Benefit Returns) were added in. A value of 1 was added to each Technology Item for every Driver upon which it had a primary effect. The Cost/Benefit Return was given a rating of high, medium or low, and equated to a weight of 2 for high, 1 for medium and 0 for low. (Note: Need dates were determined not to be significant factors for these Technology Items.)

The priority factors were then summed for each Software Technology Item to derive an overall rating as shown in the priority evaluation chart in Figure 2-4. The Technology Items identified with an asterisk are those that fall in the top third of the overall ratings, and thus are those recommended for immediate NASA consideration.

2.4 Development Schedule

A schedule for the development or advancement of technology required by each of the selected Software Technology Items is contained within each Technology Development Plan. In general, each item is given a 1978 start date, as most of them are immediately necessary with the advent of the STS operational phase in 1980 and each will require a year or more to complete.

For planning purposes it was assumed that development of an operational application would take two years after demonstration of a

PRIORITY EVALUATION

TI No.	Critical Enablement	Desirable Enablement	Cost Reduction	Enhancement	Priority	Applicable Drivers	Cost/ Benefit Return	Need Date	Rating	Top 1/3
01		X	X	X	5	3	1	N/A	9	*
02			X		3	1	0	N/A	4	
03			X		3	1	1	N/A	5	
04			X		3	1	0	N/A	4	
05			X	X	4	1	2	N/A	7	
06			X		3	1	0	N/A	4	
07			X		3	1	0	N/A	4	
08	X		X	X	10	3	2	N/A	15	*
09		X	X	X	5	2	1	N/A	8	*
10		X	X	X	5	3	1	N/A	9	*
11		X		X	2	2	0	N/A	4	
12		X			1	1	0	N/A	2	
13				X	1	1	0	N/A	2	
14		X			1	2	0	N/A	3	
15	X			X	7	1	2	N/A	10	*
16		X			1	1	0	N/A	2	
17			X	X	4	4	0	N/A	8	*
18			X		3	2	0	N/A	5	
19			X	X	4	1	1	N/A	6	
20		X	X	X	5	15	2	N/A	22	*

Figure 2-4

prototype model. For instance, an image processing application needed in 1985 would have to be proven feasible by 1983 in order to allow time for development of a flight item.

In any event, the benefits that will accrue from the development of any Technology Item cannot begin until the development is complete.

3. SOFTWARE TECHNOLOGY ITEM DEVELOPMENT PLANS

3.1 Requirements Decomposition and Structuring Guidelines (TI-01)

3.1.1 Problem Definition

Software design practice up to now has suffered from a lack of consistency among designers in their approach to decomposition of the requirements and development of design structure. The need for rational guidelines becomes apparent when one considers the fact that totally different designs can result from different designers working on identical software requirements. This makes it almost impossible to evaluate designs, much less to identify the optimum design. Unlike hardware, in which standard components are integrated by well-established logic principles, each software design is a new design. This is costly in development and implementation, but even more so in the training and maintenance requirements to support utilization phases of an operational system.

Of even more importance, lack of consistent, good design compromises the most valuable characteristic of software -- the flexibility to adapt to changing requirements. The study proposed to help solve this basic Software Engineering problem is expected to be of moderate cost (3-10 man years).

3.1.2 Technology Development Requirements

The technology developments needed under this broadbased study are the guidelines for requirements decomposition and structuring in software design. These developments must contribute towards achieving consistency in design. Also, it is necessary to derive and establish proper design evaluation criteria.

Technology must be developed which will provide a means to convey what the requirements are, and will allow the breakdown of these requirements into a functional distribution. This should lead to software designs that are reliable and easily understood, implemented, used, and maintained.

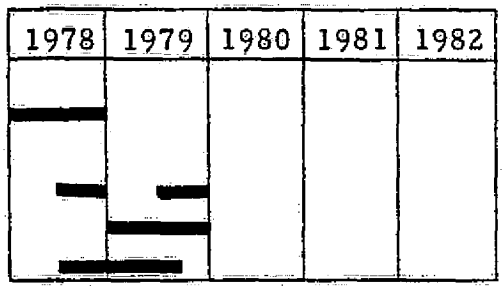
3.1.3 Suggested Approach

The study proposed for this effort has four steps as shown under the schedule. The first step is the establishment of requirements decomposition guidelines. A comprehensive set of guidelines for decomposition of the furnished software requirements is to be developed. This will be done through a thorough review of the various current approaches to the problem and a critical analysis of the relative advantages and disadvantages of these approaches. The next step is to develop a logical set of guidelines

aimed at achieving consistency of design across a given set of requirements. This must be followed by the establishment of techniques which will ensure proper structuring of functions into a viable, usable design. Development of guides to proper structure and consistent design is an iterative enhancement process, as shown in the schedule. In parallel with the previous activity, the criteria for measurement and evaluation of design products will be established. This will be of considerable significance in assessing the results of using these guidelines in NASA applications.

3.1.4 Schedule and Resources

- o Requirements Decomposition Guidelines.
- o Design Consistency Guidelines.
- o Structuring Guidelines.
- o Evaluation Criteria.



Estimated Cost of Development: $150K - 500K.

3.1.5 Verification of Results

The end product of this effort will be a set of guidelines for software design rather than a specific piece of software. The verification of results can be viewed as the assessment of benefits arising from application of these guidelines. Thus, this step in effect consists of applying these guidelines to one or more specific software design efforts to compare the experience with similar efforts not based on these guidelines. The relative merit will be assessed through application of the evaluation criteria, which in turn can be appraised through demonstration of actual program product results. The verification process itself will make a valuable contribution to the development of the Software Engineering profession.

3.1.6 Related Studies

- o Software Technology Items.
  - TI-08 Interface criteria for NASA distributed processor applications.
  - TI-09 Task control structures for distributed processor environments.

- TI-18 Guidelines for design documentation consistency.

- TI-19 Software prototyping methods.

o Software Technology Items (Secondary).

- TI-10 Program organization methods for real-time fault recovery.

- TI-15 Restructured image analysis software for parallel processing.

3.2 Requirements-to-Code Translation Aids (TI-02)

3.2.1 Problem Definition

Current practice of manual generation of codes from given software requirements is not only expensive but also error-prone. This is because, first, the software requirements are not specified in a standard format and hence the requirements decomposition is mostly adhoc. Second, the lack of consistent design guidelines makes the steps leading to the generation of codes more of an art (or lack of it!) rather than a science. Therefore, for a given set of requirements the generated code does not necessarily reflect the best code, and methods of proving the software as fault-free are still to be realized. Therefore, some means of automating the code generation process is essential to achieve a measure of error and cost control in software development. This requirement-to-code translation problem is likely to be of relatively high cost and require 10-20 man years.

3.2.2 Technology Development Requirements

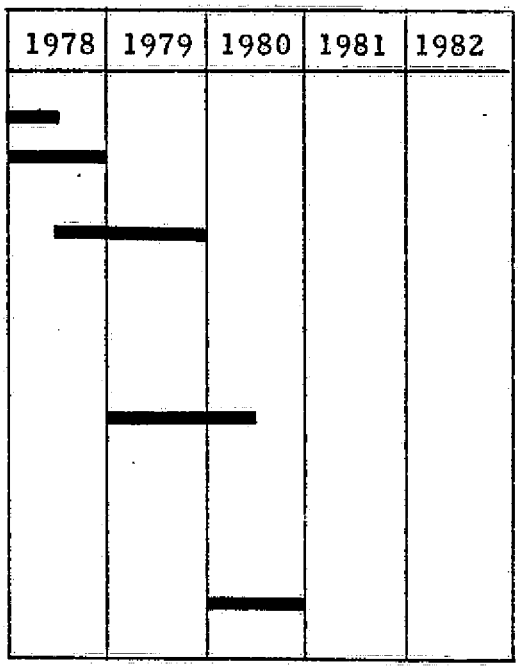
The requirements under this study are defined as the development of automated aids for generating program codes (in a desired language) from a specified set of software requirements. This technology development is tailored to meet the specific needs of payload software requirements, rather than any general study of automatic programming, to ensure direct payoff for NASA for its investment in this technology development effort.

3.2.3 Suggested Approach

The first step of this study is a review and assessment of the on-going efforts in this field as applied to other areas of software development, particularly business-oriented and other commercially funded projects. In parallel with this, a requirements analysis must be carried out to identify the spectrum of NASA payload software needs. With these preliminary efforts as the base, a master software instructions/code library, consisting of all the key functions reflecting NASA payload software requirements, can be developed. The next step is to define and develop a transformational approach which can successively transform, in stages, the input software requirements into a set of program codes using the master library as a reference. The developed methodology is then tested and the related documentation prepared to ensure its suitability for application to NASA payload software development.

3.2.4 Schedule and Resources

- Analyze requirements.
- Identify the spectrum of NASA payload software requirements.
- Develop a master software instructions code library consisting of all basic functions reflecting NASA payload software requirements.
- Develop a transformational approach which transforms the input requirements successively into a set of program codes using the library.
- Test and document the methodology developed.



Estimated cost of development: $500K - $1000K.
Resources include access to information on future payload software development plans and program details.

3.2.5 Verification of Results

The developed requirement-to-code translation aids will be test-implemented and validated as to their ability to perform the designated function. The testing will include actual application of the code generator product to a previous software implementation to derive and compare the operational capability of the automatically generated code with the earlier manually developed code.

3.2.6 Related Studies

- Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.
  - TI-03 Simplified software development system demonstration model.
  - TI-04 Query-guided implementation methods.
  - TI-18 Classification of error mechanisms.
  - TI-19 Software prototyping methods.

3.3 Simplified Development System Demonstration Model (TI-03)

3.3.1 Problem Definition

The advent of large scale usage of microprocessors in distributed systems for real-time space applications calls for simpler software development tools. This is necessary to ensure that the cost of software development will not become a barrier to the benefits of microprocessor usage which otherwise is very cost-effective.

Many software development systems exist, but it is rare to find one that is simple and even rarer to find one that is transportable. In an era of diverse users of space systems, each developing a software subset of a future integrated flight complement, ways and means of simply and economically developing software components must be found.

An effort which is designed to achieve this objective is the creation of a demonstration model of a simplified system for software development. This effort can be expected to be one of relatively high cost requiring 10-20 man years.

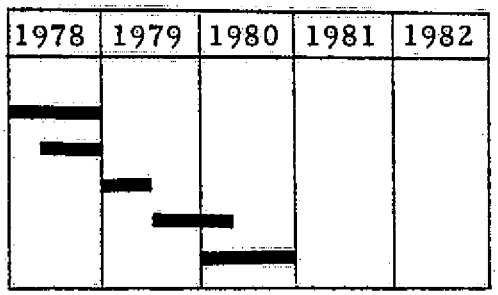
3.3.2 Technology Development Requirements

The feasibility of a simplified system for software development can be shown by the creation of an appropriate demonstration model. Accordingly, the technology requirements of the study can be stated as the building of a simplified development system demonstration model. This will be a precursor to low-cost operational systems for use by non-professional programmers in the development of space applications.

3.3.3 Suggested Approach

This development effort has several identifiable steps as shown in the schedule. The first step is the requirements definition. Requirements definition is based on developing a clear understanding of the purpose of the model and the environmental constraints under which the modeled system will have to operate. These requirements are analyzed to derive the specifications of the demonstration model. This is followed by the design of the demonstration model which will emphasize the different aspects of the operational system envisioned for the PI's use. The design phase leads to the development and test phase which in turn takes the project to the final detailed system evaluation phase. The final evaluation phase is a key element. It must be detailed enough to ensure that the follow-on work of developing an operational system can be taken up with confidence that the end product will support the anticipated heavy schedule of software development in the coming decade.

3.3.4 Schedule and Resources



- Requirements Definition.
- Requirements Analysis
- Demonstration Model System Design.
- Development and Test.
- Evaluation.

Estimated cost of development: $500K - $1000K.

3.3.5 Verification of Results

The evaluation of the demonstration model and its effectiveness is a significant part of the effort to be accomplished under this study. The schedule allows a full year for this activity and effectively portrays the importance given to this phase of the effort. This is essential to ensure the integrity of the demonstration model and the operational system to follow. Each step of the effort provides concrete output products (i.e., requirements, composition, design specifications, test system and evaluation system) for measurable verification of progress.

3.3.6 Related Studies

- Software Technology Items.
  - TI-02 Requirements-to-code translation aids.
  - TI-04 Query-guided implementation methods.
  - TI-17 Classification of error mechanisms.
  - TI-19 Software prototyping methods.

3.4 Query-Guided Implementation Methods (TI-04)

3.4.1 Problem Definition

The objective of this technology item is the development of new programming procedures wherein the programming effort is supported/guided by a questionnaire. The basic problem to be addressed is essentially similar to the one underlying TI-02, Requirements-to-Code Translation Aids, in that new automated tools for software development are called for to reduce the expensive, error prone manual approach to code generation. However, the emphasis here is less on autonomy than in the other approach and presumably the objectives are more easily realizable. This effort is consequently more moderate in cost (5-10 man years) and is expected to result in a faster return of benefits.

3.4.2 Technology Development Requirements

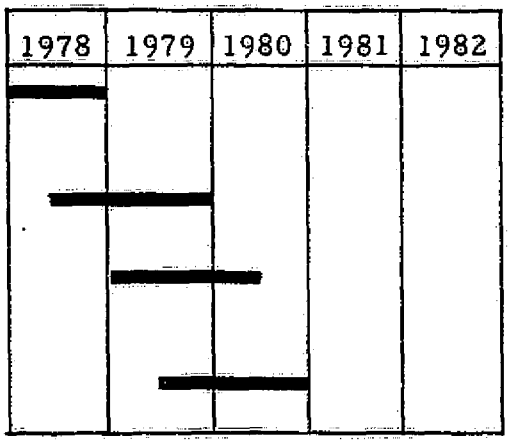
Technology development required under this study are semi-automated tools for software development using the questionnaire approach. This questionnaire approach must be especially attractive for interactive man-machine systems, operating in dynamic environments, wherein it may be necessary to create new software and/or new configurations of existing software. Query-guided implementation methods must offer the flexibility necessary for such applications. At the same time these methods must contribute towards the traditional software development goals in terms of reduced software errors and consistency in the quality of the resulting software product.

3.4.3 Suggested Approach

As a practical step, it is advisable to constrain this effort of the development of semi-automated tools to the specific NASA payload software development requirements. This limitation ensures that a viable solution will be developed within the time and cost constraints of NASA programs. Any attempt at global solutions could lead to lack of effective control over the projected costs and time schedules. The first step is, therefore, to develop and establish the common features and requirements of NASA payload software development programs. This is to be followed by an effort directed towards developing a skeleton structure of the software corresponding to established commonalities. The next step is the design and development of a questionnaire methodology which when applied to specific requirements leads to the filling out of the skeleton structure and results in the desired software package. Finally, the methodology should be thoroughly tested and documented to ensure an appropriate return on investment for NASA.

3.4.4 Schedules and Resources

- o Derive and establish the common features of NASA payload programs.
- o Develop a skeleton structure of the development system.
- o Develop a questionnaire methodology for filling out the skeleton structure.
- o Test and document the methodology.



Estimated development costs: $200K - $500K

3.4.5 Verification of Results

The end product of the study being implementable methodology, it is necessary to ensure that the developed tools can be put to effective use through testing and documentation. This testing should include test implementation in specific NASA software development programs to evaluate its effects on efficiency, reliability and cost of program development. In view of the man-machine interaction involved in questionnaire approaches, it is desirable to study programmer responses to the methodology as part of the verification process to ensure its acceptability.

3.4.6 Related Studies

- o Software Technology Items.
  - TI-02 Requirements-to-code translation aids.
  - TI-03 Simplified software development system demonstration mode.
  - TI-18 Classification of error mechanism.
  - TI-19 Software prototyping methods.

3.5 Standardization Criteria for NASA Software (TI-05)

3.5.1 Problem Definition

Lack of standardization in software design, development, testing and other facets of software projects, even within the bounded sphere of NASA activities, frequently results in expensive, often duplicated software. Needed reliability is difficult to achieve. Standardized software can make a significant contribution towards the goal of totally fault-free software. Therefore, as a first step, it is necessary to develop the criteria for standardization. The degree of the problem solution depends on the scope of the standardization process; however, as proposed here is expected to be of moderate cost involving 3-10 man years of effort.

3.5.2 Technology Development Requirements

Development of standardization criteria is the goal of this technology development item. This criteria development will span the spectrum of activities associated with a software development project. For example, standardization has many implications: standard language design, standard operating system design, and standard interface implementation. A degree of standardization in each of these areas and criteria for such standardization is desired and sought under this effort.

3.5.3 Suggested Approach

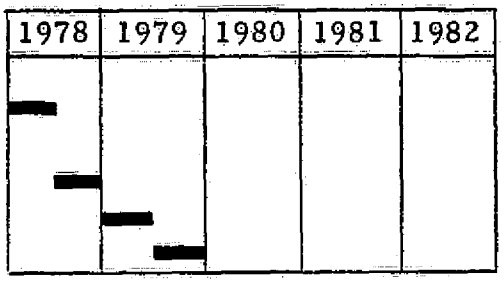
The first step must be to understand and analyze the implications of standardization. Through this effort, it should be possible to identify the specific elements of a software project which can be considered prime candidates for standardization. Once the areas deemed feasible for standardization are established, the next logical step is to develop the criteria for standardization.

It is advisable to gain some insight into industry practices in standardization and apply these ideas with necessary modifications to the NASA environment. Modification and possibly new concepts may be necessary in view of (1) the decentralized nature of the NASA organization (and the diversified contractor force supporting it), (2) the consequent problems of coordinating the software development efforts, and (3) the difficulties in enforcing the standardization procedures. Selected candidate software packages should be weighed and tested against the developed standardization criteria to validate their effectiveness in establishing standard NASA software libraries for the 1980-1990 time frames.

New developments in software design methodology which might influence the standardization criteria are conceivable and modifications at a later date may have to be made. However, irrespective of such future changes, it is necessary to have fairly rigid standardization criteria to avoid duplication of efforts and minimize the software development costs and associated errors.

3.5.4 Schedule and Resources

- o Analysis of Standardization Implications.
- o Development of Criteria.
- o Identification of Candidates.
- o Evaluation of Validity.



Estimated cost of development: $150K - $500K.
Resource requirements include NASA-wide access to software projects, programs and present practices for minimizing duplication of efforts.

3.5.5 Verification of Results

The end product of this study is a set of standardization criteria. The verification process essentially consists of applying the criteria to selected software packages and development projects to determine the validity and feasibility of enforcing such criteria in the development process.

3.5.6 Related Studies

This Technology Item is pertinent to virtually every aspect of the computer software domain.

3.6 AOL Design Guidelines/Standards (TI-06)

3.6.1 Problem Definition

Application-oriented languages (AOL) are expected to dominate the scene in future decades with users developing the software required for their particular needs. This requires appropriate AOL design guidelines and standards to ensure that the AOL's developed to meet the needs of a changing environment are compatible with NASA system goals and have the requisite language consistency. This developmental effort is expected to be one of relatively modest cost involving 2-5 man years.

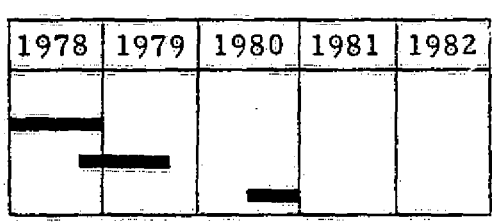
3.6.2 Technology Development Requirements

Development of design guidelines and standards for application-oriented languages (of interest to NASA applications) is the technology advancement required by this item. This requirement is the first step towards the goals of economical development of efficient application-oriented languages, particularly those of real-time onboard distributed systems. Design guidelines/standards are to ensure that the resulting AOL's are simple to use and efficient even when deployed by non-professional programmers. These requirements tie this effort to the one proposed in the next item (TI-07) on AOL compiler generator cost factors. The AOL design guidelines will determine the complexity of the compilers and therefore influence the cost of generating these compilers. Accordingly, the developments under this study have in view the objectives and associated factors of TI-07.

3.6.3 Suggested Approach

As illustrated in the schedule, the approach consists of three specific steps. The first is the determination of the concepts underlying AOL and understanding what its requirements are. These concepts are an essential prerequisite to the development of guidelines for AOL design. The next step is the actual development of guidelines to meet the requirements as derived. The third step is the formulation and development of criteria to evaluate the AOL design resulting from the use of these guidelines. This provides a means of verifying the applicability and usefulness of the derived guidelines in practical problems of AOL design.

3.6.4 Schedule and Resources



Estimated cost of development: $100K - $250K.

3.6.5 Verification of Results

The guidelines derived for the study must be applied to specific AOL design projects to test the applicability of these guidelines to real-world NASA problems. The evaluation criteria will then be deployed as the tool for assessing the resulting design, which reflects on the usefulness of the derived guidelines in providing an effective AOL to an actual user.

3.6.6 Related Studies

- Software Technology Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-07 AOL compiler generator cost factors.
  - TI-18 Guidelines for design documentation consistency.

3.7 AOL Compiler Generator Cost Factors (TI-07)

3.7.1 Problem Definition

The advent of increased development and usage of application-oriented languages (AOL) calls for design of an increasing number of compilers. Compilers being machine-dependent, the problem is further enlarged by the continuing spread in the number of different computers coming into usage. Compiler generation, however, has remained an expensive task. Thus in view of this expected increase in compiler design activities, it is necessary to look into why compilers cost so much. This study of investigating the factors contributing to the cost of compiler generation is viewed as one of relatively modest effort (2-3 man years). This will lead to methods of significantly reducing compiler generation cost.

3.7.2 Technology Development Requirements

The requirements of this task are of a different nature from the other technology items. The end product is not a specific technique or software package, but a qualitative assessment of the major factors contributing to the cost of compiler design, from which it should be possible to derive techniques for reducing these costs.

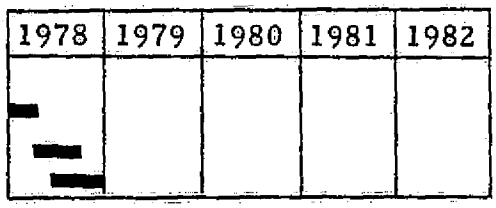
That problems exist in compiler generation is a known fact. Why they exist is another question. The requirement of this task is to provide an answer in quantifiable terms that will lead to new and feasible approaches to the development of tools to support application growth.

3.7.3 Suggested Approach

The study to be undertaken consists of three steps as shown in the schedule. The first step is to identify and typify representative AOL compiler generators used to generate payload application software. This will be followed by an in-depth analysis of the development history of these generators. The analysis must take into account the particular environmental constraints and other factors associated with the development of each of these generators. The analysis must identify the development cost and extract those factors contributing to the cost. Each of the contributing factors must be assessed in terms of significance and potential impact on future development programs. The result will be a comprehensive overall assessment of the problem of controlling costs in AOL compiler generation, and the identification of what steps must be taken to reduce future cost. This effort should be correlated with that under TI-06 to ensure that the final assessment reflects the benefits of development under TI-06.

3.7.4 Schedule and Resources

- o Identify typical AOL generators.
- o Trace development history.
- o Perform cost assessment.



Estimated cost of development: $100K - $150K.

3.7.5 Verification of Results

The end product will be a specific set of factors which will provide a tangible product in terms of the benefits accrued in future compiler generation programs. As such the test would be to apply the experience gained by the analysis of these cost factors in a test compiler generation exercise and to assess the associated costs relative to the costs of prior comparable programs.

3.7.6 Related Studies

- o Software Technology Item.
  - TI-06 AOL design guidelines/standards.

3.8 Interface Criteria for NASA-Distributed Processor Applications (TI-08)

3.8.1 Problem Definition

The expanding role of distributed processor systems envisioned for future payloads and missions demands that the associated problems of interprocess communication and related aspects be studied in depth by NASA. This problem is being addressed for onground systems within the industry. However, the special implications of onboard near-real-time processing environments need to be carefully examined by NASA. The development of suitable interface criteria for NASA-distributed processor applications is one of modest cost (2-4 man years). However, it is mandatory in terms of integrating and absorbing the multiplicity of high rate data sources projected for the next decade.

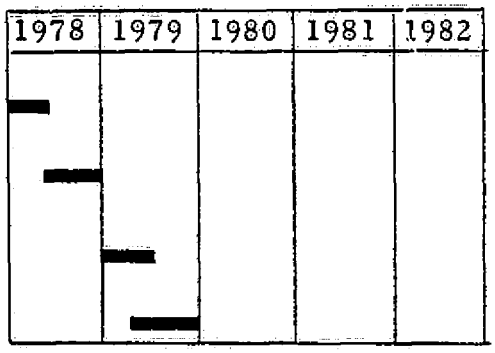
3.8.2 Technology Development Requirements

Interface criteria to assist in the design of NASA-distributed processor interfaces represent the major end products expected of this technology development effort. The criteria are necessary to simplify problems of integration and to minimize associated costs by avoiding recourse to nonstandard interfaces. These developments are critical to increased and cost-effective utilization of distributed processor concepts and techniques, and are essential to the enablement of intelligent instruments.

3.8.3 Suggested Approach

The first significant step of this effort is the identification of the potential distributed system configurations for deployment in near-real-time spaceborne environments. Once these are identified, it is necessary to analyze in detail each of these configurations to determine parametric characteristics such as data flow, and to establish appropriate control structures. These parameters characterize the interprocess communication needs. Based on these quantitatively established needs, the appropriate interprocess communication techniques can be selected through trade studies. This will in turn lead to the interface criteria which are essential to widespread deployment of distributed processor systems within the NASA programs.

3.8.4 Schedule and Resources



- o Identify potential distributed configurations.
- o Characterize data and control flow.
- o Select interprocess communication techniques.
- o Establish interface criteria.

Estimated cost of developments: $100K - $200K.

3.8.5 Verification of Results

The end product is a set of interface criteria. The test of its validity lies in its application to real-world problems, and verification can be achieved through test implementation of selected cases.

3.8.6 Related Studies

- o Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.
  - TI-05 Standardization criteria for NASA software.
  - TI-09 Task control structures for distributed processor environments.
  - TI-10 Program organization methods for real-time fault recovery.
  - TI-15 Restructured image analysis software for parallel processing.

3.9 Task Control Structures for Distributed Processor Environments (TI-09)

3.9.1 Problem Definition

Task allocation and control is major significance in the design and operation of ground-based distributed systems and has received considerable attention. This function is much more critical in the management of real-time space-based distributed systems and it is decidedly in the NASA domain to tackle this problem. The task allocation process should ensure that the available resources are utilized to the fullest extent at all times and as efficiently as possible, given the particular tasks and processor configuration. Once the optimal task allocation is determined, then the actual task control should be implemented in accordance with the planned allocation. Development of the task control structure represents the other major part of the problem contained in this item. It is expected that, in view of the progress in ground-based systems, this effort will be of relatively low cost and require 2-5 man years of effort.

3.9.2 Technology Development Requirements

The technology development requirements are twofold: (1) techniques for automatically determining the optimal task allocation and scheduling among the processors within the distributed system, and (2) design and development of control structures for enforcing the optimal task allocation and scheduling. These tasks are interrelated and should be carried out with close coordination between the two efforts. This item has particularly great applicability to the effort proposed under TI-15 (restructuring image analysis software for a parallel processing). The technology development efforts under this plan should therefore complement the other effort.

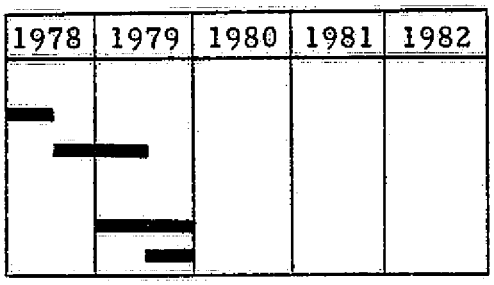
3.9.3 Suggested Approach

The starting point for this plan is to clearly identify the scope and type of tasks as well as the nature of the distributed system environments expected to arise in the implementation of planned NASA missions. Concurrently, available techniques for onground distributed systems should be reviewed to delineate the deficiencies of such methods in meeting the requirements of real-time space systems. The two efforts together will provide the technical base for initiating the development of appropriate task allocation and scheduling techniques. These techniques should be amenable to automated implementation. This will lead to optimal allocation and scheduling of the tasks for any given processing problem

within the scope of the study. Next will be the development of suitable software control structures capable of implementing the optimal allocation and scheduling plans. The final step is testing of these techniques and control structures, through simulation of the onboard processing environments, to verify their reliability and efficiency.

3.9.4 Schedule and Resources

- Requirements analysis.
- Develop optimal task allocation and scheduling methods.
- Develop task control structures.
- Test and documentation.



Estimated cost of development: $100K - $250K.
Resource requirements include computer facilities for simulation and testing.

3.9.5 Verification of Results

The results in this case are the techniques for determining the optimal task allocation and scheduling and the control structures for their implementation. The verification process should therefore include simulation of the onboard real-time processing environment and testing of the developed techniques and control structures in such simulated environments. This will ensure that the resultant products of the study will meet the NASA requirements for distributed systems. It will also enable operational implementation of related technology developments (particularly the implementation of parallel processable image analysis functions: TI-15).

3.9.6 Related Studies

- Software Technology Items.
  - TI-01 Decomposition and structuring guidelines.
  - TI-05 Standardization criteria for NASA software.
  - TI-08 Interface criteria for NASA distributed processor applications.
  - TI-10 Program organization methods for real time fault recovery.

- TI-15 Restructured image processing software for parallel processing.

- TI-19 Software prototyping methods.

3.10 Program Organization Methods for Real-Time Fault Recovery (TI-10)

3.10.1 Problem Definition

Software and hardware errors as well as data anomalies do occur in spite of the best of precautions in the design and development processes. It is therefore necessary that space systems should be backed by effective means of fault recovery. In view of the dynamic nature of the system environment, it is also crucial that this recovery be in near-real-time. Thus, the problem addressed in this study is the need for developing efficient techniques for real-time recovery from errors caused either by hardware, data, or software. The recovery procedures to be considered here are primarily limited to software rather than to purely hardware-oriented solutions such as redundant hardware. The proposed effort is of moderate to high cost (between 4-20 man years) depending on the depth to which the problem is to be explored.

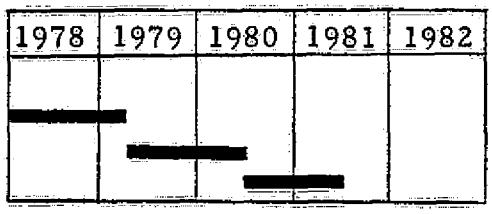
3.10.2 Technology Development Requirements

This effort includes review/development of recovery techniques most suited for NASA real-time space applications. This will be based on research into the structure and organization of software systems which provide reliable fault recovery through proper program organization methods. These developments should be coordinated with the activities proposed in other technology items, such as classification of error mechanisms (TI-17).

3.10.3 Suggested Approach

Structure analysis represents the first step of the proposed effort. This consists of identification of the types of faults likely to be encountered in real-time space systems and determination of the nature of the recovery process required to counteract these faults. Development of appropriate techniques and procedures for system recovery is the next step. This is followed by development of program organization methods which will maintain the status of the system so that the system can recover gracefully with minimum loss of data and control.

3.10.4 Schedule and Resources



- o Real-time program structure analysis.
- o Recovery techniques.
- o Program organization methods.

Estimated cost of development: $200K - $1000K.

3.10.5 Verification of Results

The development methods should be tested and evaluated by simulation under appropriate fault-inducing environments. This is extremely essential to prove the reliability of the techniques for deployment in real-world applications wherein the risk factors associated with improper recovery cannot be tolerated.

3.10.6 Related Studies

- Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.
  - TI-08 Interface criteria for NASA-distributed processor applications.
  - TI-09 Task control structures for distributed processor environments.
  - TI-12 Control structures for adaptive systems.
  - TI-15 Restructured image analysis software for parallel processing.
  - TI-17 Classification of error mechanisms.

3.11 Adaptive High-Speed Buffering Techniques for Dynamic Networks (TI-11)

3.11.1 Problem Definition

Missions involving image acquisition and other high-rate data-generating instruments, call for advanced capability in all areas of the data acquisition, handling, and processing system. The variable data rates of different experiments and the need for intelligent data acquisition dictates that advanced techniques, capable of adapting to changing data environments, be developed. High-speed buffering techniques in dynamic acquisition and switching networks are of prime importance, particularly in view of the high-speed, large-scale memory capability expected in the near future. Methods for optimal utilization of potential buffer resources will have tremendous impact on information gathering systems. This study represents a moderate to high cost developmental effort requiring 5-15 man years.

3.11.2 Technology Development Requirements

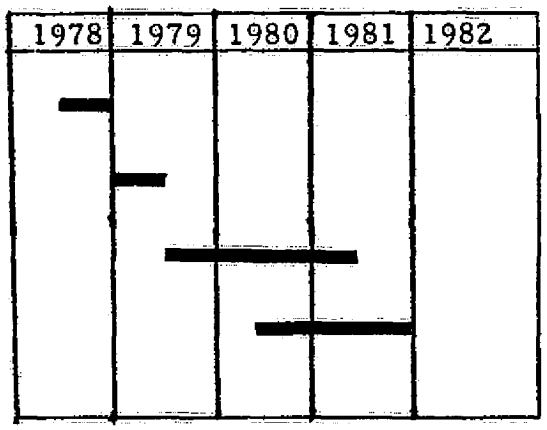
The technology development required by this item is the development of advanced high-speed buffering techniques capable of adapting to the dynamic characteristics of the data environment envisioned for future missions. These missions will involve high data rate experiments such as multispectral remote sensing of earth resources. This development is required to ensure that there will be an optimal utilization of the of the resources available both onboard and elsewhere in the total data system. This effort will contribute towards the overall objective of data system optimization by upgrading the state of the art in the key area of buffering techniques.

3.11.3 Suggested Approach

This developmental effort consists of several distinct steps as indicated under the schedule. The first step is a detailed assessment of the existing technology to identify deficiencies in the state of the art in meeting the future requirements. This requirements analysis must be supported by a parallel effort to identify the possible advancements expected in related hardware/software technology areas. With these efforts as the technology base, the study will proceed with the main task of developing new adaptive techniques. The possibilities of incorporating at least some basic level of intelligence at the front end will be explored. For example, when one is only interested in change detection, only data that are subject to change need be stored. The developed techniques will then be implemented, tested, and documented to ensure a sound return for NASA investment.

3.11.4 Schedule and Resources

- o Survey available techniques.
- o Identify advancements in related areas.
- o Develop new adaptive techniques.
- o Implement, test and document these techniques.



Estimated cost of development: $250K-750K.

3.11.5 Verification of Results

The developed techniques will be implemented and tested under properly simulated data environments including the expected range of dynamic data characteristics. This requires a thorough knowledge of the state of the art in data system simulation techniques in addition to those needed in the development phase of the effort. This verification of results, in terms of capability for meeting real-time requirements and other needs of the data system, will serve to enhance the value of the effort and help its acceptance by the user community.

3.11.6 Related Studies

- o Software Technology Items.
  - TI-19 Software prototyping methods.

3.12 Control Structures for Adaptive Systems (TI-12)

3.12.1 Problem Definition

Automated intelligence activities involve a variety of scene analysis, image processing and pattern recognition functions. These functions have to be called upon for processing, depending on the environmental conditions, in different sequences and configurations. This dependency requires the system to be adaptive to the needs of the changing environment. The problem is therefore to provide adaptive framework for the dynamic selection of the many software functions required in autonomous missions. The development of appropriate control structure concepts is expected to be of moderate cost requiring 8-12 man years.

3.12.2 Technology Development Requirements

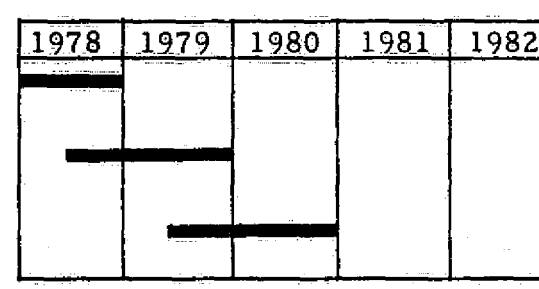
The technology advancement required is the development of control structure concepts for adaptive control of software procedures. The desired software system requires certain cognitive and decision making capabilities to adaptively deploy the right processing modules in an autonomously operating environment. The core development required encompasses conceptual, algorithmic and implementational aspects of the problem.

3.12.3 Suggested Approach

The initial step is to establish the requirements as stated above in more precise terms. A requirements analysis includes a review of the different processing functions involved in autonomous environments and their interrelationships. Once the specific processing functions and the possible configurations of their deployment are identified, it is then necessary to develop the concepts and techniques for adaptive control of these functions which can lead to optimal performance of the software system. This development of appropriate control structures will be followed up by experimental simulation (to check out the control structures under different test environments) and documentation to ensure its utility.

3.12.4 Schedule and Resources

- o Identify the interdependencies and relationships of software functions.
- o Develop appropriate control structures.
- o Test and document the developed control structures.



Estimated cost of development: $300K - $500K
Resource requirements include access to a computer system for test simulation of the developed control structures.

3.12.5 Verification of Results

A detailed simulation study plan must be formulated and implemented to ensure that the developed procedures meet their requirements. The simulation process should be carefully designed to provide an opportunity to test all feasible configuration options expected of the final software package.

3.12.6 Related Studies

- Software Technology Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-14 Adaptive search and sort routines.
  - TI-15 Restructured image analysis software parallel processing.
  - TI-19 Software prototyping methods.

3.13 Impacts of Natural Communication Methods on NASA Payload Systems (TI-13)

3.13.1 Problem Definition

Multi-experiment missions call for a more optimal utilization of human resources. The control and monitoring of the different experiments should be spread, to the extent feasible, over all the different human faculties capable of interaction and communication. The computer system in control of these experiments must provide for man-machine communications and should make the best use of all known natural communication methods. The problem is the need to extend the scope of methods used onboard for man-machine interaction. This effort is expected to be one of relatively modest cost involving 3-6 man years.

3.13.2 Technology Development Requirements

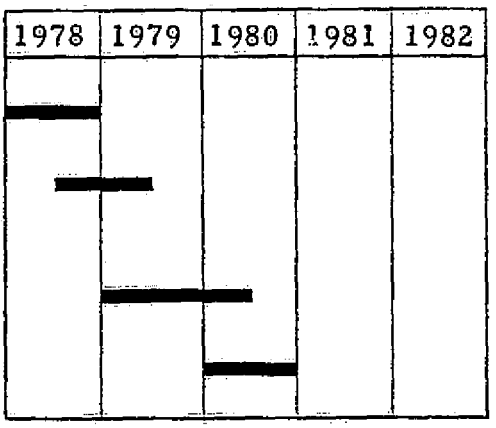
The goal of natural communication methods is best furthered by a study geared to assess their impact on the design of payload software systems. The assessment process must necessarily include a detailed analysis of typical implementations of such methods to develop a thorough understanding of existing potential. The study must critically evaluate the technology associated with these methods to clearly identify the feasible or potentially applicable methods which meet the NASA requirements for onboard implementation.

3.13.3 Suggested Approach

The starting point of this effort is a requirements analysis phase wherein the NASA requirements for natural communication methods in the upcoming missions are to be clearly identified and firmly established. Typical cases will be analyzed in depth through implementation/simulation to derive an understanding of their influence and potential impact on the supporting payload software system. This will lead not only to an evaluation of the feasibility of using these natural communication methods onboard, but also to an assessment of the necessary and desirable modifications to payload software system design. The results of such test case studies are then reviewed to derive an overall assessment of natural communication methods on space systems.

3.13.4 Schedule and Resources

- Identify NASA requirements for natural communication methods.
- Study the impact of these methods through analysis of typical implementations.
- Delineate necessary modifications to payload software system.
- Obtain an overall assessment of natural communication methods.



Estimated development costs: $150K - $250K.
Resources include the hardware for test implementation in a simulated onboard processing environment.

3.13.5 Verification of Results

The feasibility of employing the different natural communications methods will be verified by actual implementation in a simulated onboard processing environment. Such implementation will additionally lead to testing the supporting software system, thus enabling useful and reliable insight into all the different facets associated with natural communication techniques.

3.13.6 Related Studies

- Software Technology Items.
  - TI-08 Interface criteria for NASA-distributed processor applications.

3.14 Adaptive Search and Sort Routines (TI-14)

3.14.1 Problem Definition

Many of the pattern classification and image analysis techniques basic to an image processing system involve table look-up procedures operating on large arrays. These procedures necessarily call for search and sort operations in both the table formulation and look-up phases. The process is repeated innumerable times, especially in the look-up phase. This is because of the extensive volume of data handled by missions such as those involving earth resources survey and similar large imagery data acquisition experiments. It is therefore necessary to make such repeated table look-up operations as efficient as possible if images are to be processed onboard. Therefore, more efficient search/sort procedures are needed.

The problem facing the study is the need to overcome the possible inadequacies of presently known methodology in meeting the future onboard real-time processing system requirements. Further, it is desirable for these procedures to be adaptive to the array data characteristics to ensure optimum performance. It is expected that the required study is of relatively modest cost and will be approximately 3-6 man years of effort. This is because of the significant amount of related work reported in the area of search and sort techniques as applied to ground environments.

3.14.2 Technology Development Requirements

The technology development requirements can be precisely stated as the development of a set of search and sort routines and associated software control structures for adaptively selecting the appropriate routines for different data environments. This requires identification of the most appropriate methods for different array data characteristics. This will be done through review and modification of existing methods and/or additional developments followed by integration of these procedures within an adaptive structure.

3.14.3 Suggested Approach

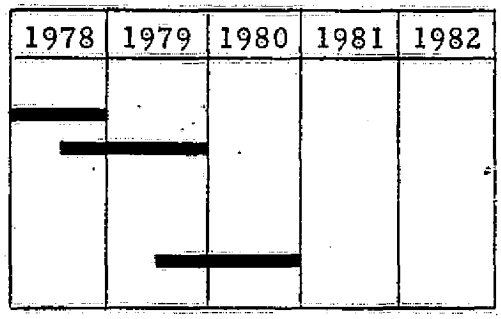
The first step of this development study is a detailed requirements analysis. This consists of a close scrutiny of the major pattern classification and image processing techniques to identify the specific needs for efficient search/sort procedures. These needs must be analyzed in the context of available information about the data rates, image sizes and such other related items pertaining to the data environment expected in future missions. This leads to a detailed definition of the requirements for

search/sort routines under different identifiable classes of data environment characteristics as well as associated time constraints for possible real-time implementation.

The next step is to evaluate the available search/sort procedures in terms of their ability to meet these requirements and identify the potential candidates for adaptation. These candidate techniques are modified, improved and combined to enhance their potential. This effort may include the development of completely new methodology. However, such a need may not arise in view of the extensive work being performed on ground-based systems. The selected set of modified/developed techniques deemed optimal for the different data environments are then integrated into appropriate software control structures. These software control structures will be able to recognize the data environment characteristics and route the data flow through the appropriate software modules. The developed package will then be tested in simulated data environments to verify and validate the end product of the study.

3.14.4 Schedule and Resources

- Requirements analysis.
- Develop adaptive routines and integrate them through appropriate control structures.
- Implement, test and document the adaptive routines package.



Estimated development costs: $150K - $250K.
Resource requirements include access to a computer system capable of simulating the different data environments expected onboard.

3.14.5 Verification of Results

The developed set of procedures and their adaptive capabilities must be verified by simulation of the different data environments. This calls for appropriate simulation tools including a computer system of sufficient computational power. The verification process should be performed both at the level of individual routines to assess individual performances and at the integrated level to verify the adaptation capabilities of the total system.

3.14.6 Related Studies

- Software Technology Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-12 Control structures for adaptive systems.
  - TI-15 Restructured image analysis software for parallel processing.
  - TI-16 Optimal large array partitioning procedures.
  - TI-19 Software prototyping methods.

3.15 Restructured Image Analysis Software for Parallel Processing (TI-15)

3.15.1 Problem Definition

Parallel processing is an attractive concept for the efficient realization of image processing and pattern recognition functions onboard NASA payloads in the coming decades. However, existing software techniques in the image analysis area are structured only for batch processing in conventional sequential processing environments on ground. Therefore, image analysis software must be appropriately tailored for implementation in parallel processing environments. The effort required for meeting the needs of this concept is expected to be moderately large in scope and to span 10-25 man years, depending on the number of software functions to be covered by the effort.

3.15.2 Technology Development Requirements

The enabling of onboard near-real-time image processing and pattern recognition activities requires full utilization of parallel processing technology to minimize the processing time required. This calls for operationally efficient software, structured to take advantage of the parallel processing environments feasible onboard in the next decade. Hence the major technology needed is the restructuring of image analysis software which is considered crucial to the enablement of image processing and pattern recognition activities onboard.

3.15.3 Suggested Approach

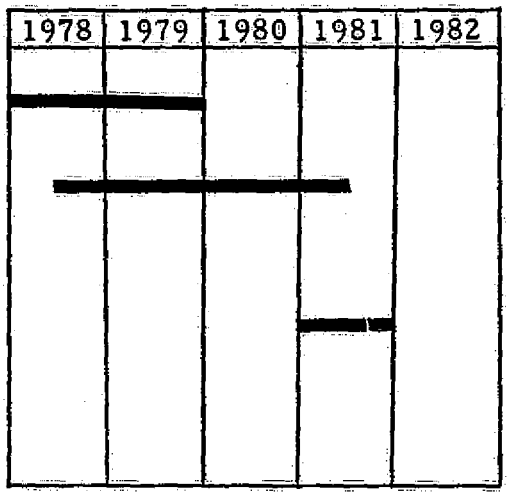
Parallelism can be visualized in terms of performing identical processing on different segments of an image or data array using parallel processing elements. It can also be visualized in the sense of carrying out independent computations on the same segment of data concurrently through different processing elements. It is therefore necessary to explore all such feasible parallelism in the major image processing functions of consequence to onboard implementation. For this purpose, it is also necessary to assess the state of the art in parallel processing techniques and supportive low cost hardware technology. This will help in determining the level or extent of parallelism advisable from a practical viewpoint.

Having identified the feasible and practical parallel processable functions and the key techniques, the software is accordingly restructured to achieve the maximum parallel processability. This is followed by identifying the appropriate computer architecture and control structure for implementing the restructured software system. In this context, the

impact of developments in the area of optical and hybrid (optical and digital) processing should be assessed to ensure that the restructured system does not attempt to perform functions that are more efficient in optical processing techniques. This assessment must include the complexities of transforming images between digital to optical domains as that tends to offset the advantages of optical processing.

3.15.4 Schedule and Resources

- Identify parallel processable functions.
- Restructure/modify the software to achieve maximum parallel processability.
- Identify the appropriate computer architecture for implementing the restructured software system.



Estimated development costs: $500K-$1000K

Resources include hardware for testing the restructured software in appropriate parallel processing environment.

3.15.5 Verification of Results

The results are to be verified through tests involving classes of multispectral data. Experiments should be performed with both the original and restructured software (the latter in parallel processing environment) to verify the correctness of the restructured software and to determine the computational gains achieved through parallel processing. This aids in verifying and validating the computer architecture and control structures to be used for purposes of onboard image processing.

3.15.6 Related Studies

- Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.

- TI-05 Standardization criteria for NASA software.

- TI-08 Interfacing criteria for NASA distributed processor applications.

- TI-09 Task control structures for distributed processor environments.

- TI-12 control structures for adaptive systems.

- TI-20 Software prototyping methods.

3.16 Optimal Large Array Partitioning Procedures (TI-16)

3.16.1 Problem Definition

Manipulations of large two- and three-dimensional arrays which arise in the implementation of many data compression and image processing functions pose an operational problem that is far from trivial. Quite often the transposition of a matrix which is too large to fit in core, may require more time than performing basic transform or other operations on the rows and columns of this matrix. In addition, the complexity of a large number of input/output operations demands more efficiency in manipulating large arrays under the constraints of a real-time image analysis system.

This problem was identified in Technology Driver PS-15, Efficient Large Array Manipulation Procedures. A solution is expected to result from a study effort of relatively modest cost (3-6 man years).

3.16.2 Technology Development Requirements

The technology developments required to solve the problem defined above can be summarized as the development of optimal large array partioning procedures. The resulting procedures should be capable of adaptating themselves to the array data characteristics, i.e., the actual procedure used must be tailored to the array data characteristics expected in the corresponding environment. Currently known and reported methodologies for performing such matrix transpositions must be evaluated to pinpoint deficiencies in meeting the requirements of onboard real-time image processors. Partitioning of the arrays is a feasible approach in meeting the constraints expected. It is necessary to develop partitioning procedures which make the best use of the available computational resources and at the same time minimize time utilization. This technology development program shall study the impact of advancements in hardware technology such as electronic cyclic memories. This will ensure that the developed procedures are consonant with the state of the art in related fields, and that the resulting methodologies will meet the NASA requirements in the most efficient manner.

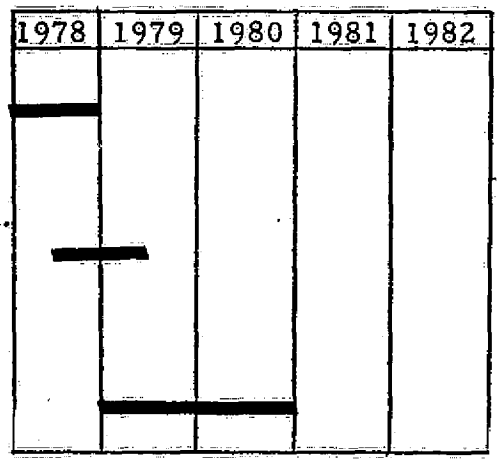
3.16.3 Suggested Approach

First, the study should consider the onboard processing system architecture (and constraints) expected by mid 1980's, and should restructure the optimal control problem accordingly. Second, alternatives to the dynamic programming approach should be explored.

Third, the feasibility of adapting the procedures to array data characteristics should be investigated. Fourth, the impact of technological advancements such as electronic cyclic memories on the manipulation procedures should be studied. The study should then be directed towards the development of suitable optimal partitioning procedures which can be applied to a variety of onboard processing environments. The deliverable end product items will be a set of strategies or optimal large array partitioning procedures which will lead to best utilization of the available onboard computational resources. In addition these procedures shall be adaptive to the array data characteristics.

3.16.4 Schedule and Resources

- Study feasibility for adapting array manipulation procedures to array data characteristics.
- Determine interrelationships of these procedures with data storage and processor architecture.
- Develop, test and document these procedures.



Estimated development costs: $150K-250K. Resource requirements include access to a computer system capable of simulating different onboard processor environments.

3.16.5 Verification of Results

The methodology developed must be tested in different simulated onboard environments on a ground-based computer system to validate and verify optimality. The tests should be designed to cover a wide variety of array data characteristics to prove the adaptability of the resulting procedures to such characteristics. These tests will ensure the success of the task undertaken and can lead to advancement of the technology to the next level required for actual onboard implementation.

3.16.6 Related Studies

- Software Technology Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-12 Control structure for adaptive system.
  - TI-14 Adaptive search and sort routines.

3.17 Classification of Error Mechanisms (TI-17)

3.17.1 Problem Definition

"Learning from History" is the most often suggested route for avoiding repetition of previously encountered errors. This is particularly valid in the case of software errors because of the lack of standardization in software design procedures and the consequent tendency to fall into the same errors repeatedly. The problem is therefore the need for development of avoidance techniques through a process of identification and classification of error mechanisms recorded in past software projects. NASA experience in space systems is particularly applicable. The effort required here is approximately 3-6 man years. The return, although perhaps difficult to measure, can be substantial in terms of cost avoidance.

3.17.2 Technology Development Requirements

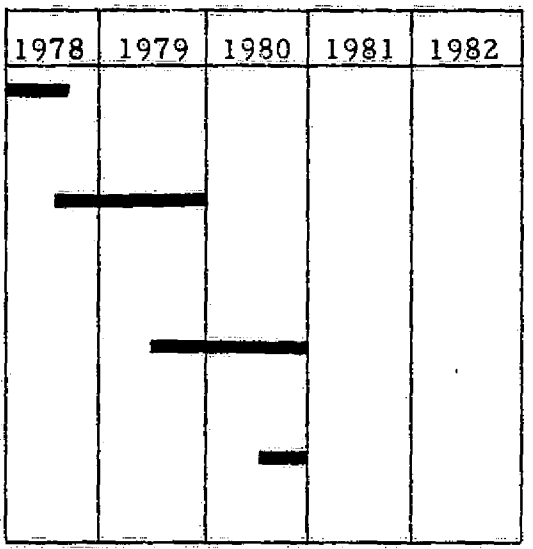
The technology developments required here are quite unlike others in that the end product is not so much a methodology but a directory categorizing the different types of software errors and the factors which lead to them. This categorization or classification of error mechanisms will be derived through analysis of prevalent errors encountered in prior space software development programs. Guidelines will thereby be developed to ensure that those pursuing future software development will be forewarned of potential faults.

3.17.3 Suggested Approach

The first step of the study is the compilation of errors documented under the different NASA software development projects of the past space programs. (This requires free access to such information by the organization conducting the study.) The compiled list should then be critically reviewed and grouped on the basis of common characteristics. An understanding of the causes of such errors will aid in developing classification of the error mechanisms. Guidelines will be developed to avoid these types of software errors in future design projects. This approach can be expected to enhance the search for fault-free software development in future NASA programs.

3.17.4 Schedule and Resources

- o Compile error data of past NASA software program projects.
- o Review and study the errors to extract their characteristics and model these errors into identifiable classes.
- o Derive an overall comprehensive list of these error classes.
- o Develop guidelines for avoiding such errors in future projects.



Estimated cost of development $150K - $250K.
Resources include unrestricted access to documented reports of experience of prior NASA software development programs.

3.17.5 Verification of Results

The concept of verification of results is not valid in the usual sense of the phrase as only future experience can prove the value of the guidelines developed from this process of classification of error mechanisms. However, a small scale software development program can be carried through in the light of these results and error avoidance costs estimated in a subjective manner. As in many research items, insight into the underlying causes of an effect can lead to unrelated advancements in technology.

3.17.6 Related Studies

This Technology Item relates to all software development technology areas.

3.18 Guidelines for Design Documentation Consistency (TI-18)

3.18.1 Problem Definition

An important aspect of software design process, which has received much attention, but few solutions, is the method of documentation of software design. No satisfactory method has been developed of conveying requirements into design representations which are understandable throughout the evolving responsibilities of a program life cycle. Attempts have been made to use data flow diagrams, functional representations, networks and other means. Though each may have specific advantages, none satisfies the need for unique clarity and understandability in a universal sense.

Inadequate details, lack of consistency and such other drawbacks in the documentation of the design can cause misinterpretation of the intent of the software designer by the programmer in the coding phase. This results in error-prone software with low reliability and consequent high costs in its maintenance. It is therefore clear that the lack of proper guidelines for design documentation represents a problem of sufficient importance to be addressed by NASA to ensure the developed software is space-worthy. The effort required to meet this problem effectively is expected to be one of relatively modest cost involving 2-5 man years.

3.18.2 Technology Development Requirements

The developments expected of this study are the guidelines for consistency in design documentation which can facilitate clear understanding of the design coding requirements. This will enhance the scope for higher reliability in the generated code, thereby reducing maintenance. This is a desirable objective in space-related environments of concern to NASA. The guidelines should be broad enough in scope to cover the spectrum of NASA payload software activities and specific enough to be directly applied to the software development projects in the coming decade.

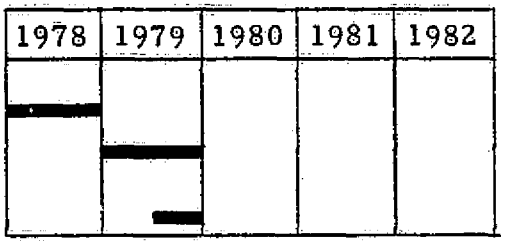
3.18.3 Suggested Approach

The effort should be closely coordinated with other software design-related Technology Items such as requirements decomposition and structuring guidelines, and AOL design guidelines/standards. The first step of this effort is to develop a clear understanding of the state-of-the-art design techniques (and hence the need to coordinate with efforts leading to improved design techniques). This will ensure that the design structure and all associated aspects are known and their implications are completely grasped prior to the development of documentation guidelines. This design structure analysis will be followed by formulation of the actual guidelines for achieving

consistency in design documentation. The next step is the development of evaluation criteria to measure the effectiveness of these guidelines.

3.18.4 Schedule and Resources

- o Design structure analysis.
- o Development of documentation guidelines.
- o Evaluation criteria.



Estimated cost of development: $100K - $250K.

3.18.5 Verification of Results

The guidelines can be effectively verified as to their worth by application to real-world software design documentation efforts. The resulting documentation is assessed on the basis of the evaluation criteria developed as part of the study. This leads to a measure of effectiveness in achieving design documentation consistency.

3.18.6 Related Studies

- o Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.
  - TI-05 Standardization criteria for NASA software.
  - TI-06 AOL design guidelines/standards.

3.19 Software Prototyping Methods (TI-19)

3.19.1 Problem Definition

Software design errors and the associated costs are always a matter of grave concern to software development projects. Even more disconcerting is the discovery that a design will not meet requirements after implementation is complete. It is therefore highly desirable to develop methods for proving the software design before committing the design to code. This is the objective of software prototyping methods. Software prototyping will provide low-cost pretesting of the key elements of the design prior to implementation of the operational system. This technology when properly developed and used offers great benefits in reliability and cost avoidance. The development effort is expected to be one of relatively modest cost involving 2-5 man years.

3.19.2 Technology Development Requirements

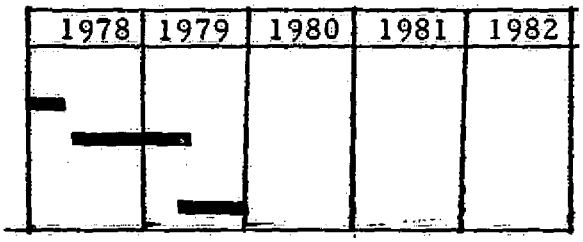
The technology to be developed under this study is the means of testing software design prior to implementation of the system. Such development will cut the high cost associated with design errors both in terms of their potential effects on reliability and in terms of the actual expenses involved in correcting these errors or making false starts. The methods to be used to achieve software prototyping capabilities represents the major requirement of this development study. Also, suitable evaluation criteria must be developed to assess the effectiveness of these methods.

3.19.3 Suggested Approach

The effort must be mainly directed towards the development of methods suitable for prototyping real-time space systems. It is necessary to assess the technology currently available in the area of ground-based prototyping techniques prior to exploring the scope for prototyping onboard real-time systems. The current technology analysis will then provide pointers for extrapolation of these techniques into the domain of space. This extension may not be straightforward in that the environmental constraints associated with space systems are likely to be of an entirely different category. The final stage of the effort is the development of appropriate evaluation criteria for assessing the developed techniques.

3.19.4 Schedule and Resources

- o Technology analysis.
- o Low-cost prototyping methods.
- o Evaluation criteria.



Estimated cost of development: $100K-250K.

3.19.5 Verification of Results

The end product of this effort and its effectiveness can be judged by application to specific cases of software design and development programs. The evaluation criteria developed as part of this technology development study will be used to assess the gains and verify the efficiency of the tools resulting from the study. Comparative analysis with non-prototype development will be performed where systems with similar requirements can be identified.

3.19.6 Related Studies

This Technology Item is applicable to all software development technology.

3.20 Software Technology Monitoring (TI-20)

3.20.1 Problem Definition

NASA should capitalize on the continuous evolution and the occasional breakthroughs occurring in the areas of hardware/software technology. A process of early awareness of the developments is needed followed by a study of the impacts of such developments on NASA programs. This requires that there be a constant monitoring of the developments in the field and for this purpose it is necessary that NASA establish a suitable formal mechanism.

As an Item, software technology monitoring seems obvious and deceptively simple. NASA personnel are continually monitoring technology and providing much of the advancement. However, this monitoring basically applies to a specific area of interest of the person or group doing the monitoring. The key to this task is to have a central source monitoring all software/hardware technology and correlating advancements to ongoing or proposed space activities wherever they may be applicable. This is to be an on-going low level effort of relatively low cost, requiring 1 man year per year.

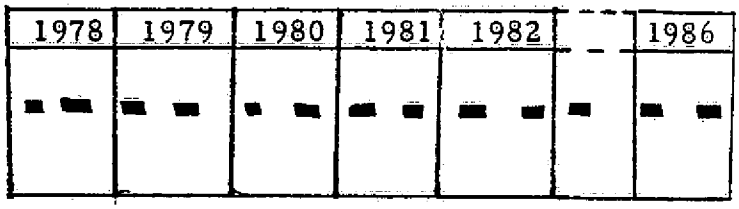
3.20.2 Technology Development Requirements

In this case, the requirement is not truly one of technology development but rather one of technology assessment. However, the requirements can be viewed as developing and instituting a formal software technology monitoring mechanism. Such a mechanism would be responsible for performing the monitoring activity and correlating developments with the NASA requirements. It would thereby contribute towards identification of additions (and/or deletions) to the NASA Software Technology Item roster.

3.20.3 Suggested Approach

The formal mechanism set up for this purpose should have access to information about activities from different sources within NASA and outside of it also, to the extent feasible. This is in addition to the open literature publications which are easily accessible. The formal mechanism should utilize the modern techniques of technology forecasting and assessment, such as advanced DELPHI techniques, not only for monitoring but also to identify the expected course of developments. This activity being an on-going effort, the projections can be verified whenever promising technology is identified.

3.20.4 Schedule and Resources



Estimated development costs: $50K/year.

3.20.5 Verification of Results

This Technology Item is significantly different from others; therefore the concept of verification results does not apply in its usual sense. However, the projections on future developments made earlier can be verified by the monitoring process, thereby refining the assessment methodology.

3.20.6 Related Studies

Not applicable (except in the sense that the Technology Item is related to the total area of software technology).

APPENDIX A

SOFTWARE TECHNOLOGY DRIVER DESCRIPTION

SUMMARY

This appendix serves to provide a summary description of each of the fifteen technology drivers (listed below) derived from an analysis of the technology emphasis areas during Phase 1 of this study and described in the Mid-Term Report (No. 76-0084), NAS8-32047, December 24, 1976, by M&S Computing, Inc.

Sl. No.	Technology Driver Title
PS-01	Software Design Engineering
PS-02	Trend Toward Software Development by Non-Programmers
PS-03	Fault-Free Software
PS-04	AOL Design Methodology
PS-05	Low Cost Development of AOL Compilers
PS-06	(Distributed) System Partitioning/Interconnection Techniques
PS-07	Very Large Storage Access Simplification
PS-08	Software Fault Detection
PS-09	Software Recovery (after fault detection)
PS-10	High Speed Buffering Techniques
PS-11	Design and Control of Adaptive Software Procedures
PS-12	Natural Communication Methods
PS-13	Efficient Large Array Search and Sort Procedures
PS-14	Parallel Processing Techniques
PS-15	Efficient Large Array Manipulation Procedures

SOFTWARE DESIGN ENGINEERING (PS-01)

A review of software development technology shows that hardly any of the facets of software development are being satisfactorily performed at the present time. Furtherance of the state of the art in all these facets (except perhaps in the actual coding phase) is not only desirable but also essential to ensure meeting the NASA payload software requirements. While each facet in itself can be viewed as a Technology Driver, it is possible to identify a basic common denominator contributing to the deficiencies in these areas. This key feature is the software design engineering. Software design engineering embraces all the different stages from an understanding and interpretation of the user-specified requirements through the documentation of the design. Some of the aspects to be considered are visibility of design, traceability, requirements documentation, design techniques, structuring, consistency, and design documentation. The entire spectrum of requirements analysis and design process is thus covered by this Technology Driver. Software costs and reliability goals will be met only when software design engineering becomes a reality. This is particularly important in the context of NASA payload software wherein the software requirements are more difficult to define and reliability considerations are of utmost significance.

TREND TOWARDS SOFTWARE DEVELOPMENT BY NON-PROGRAMMERS (PS-02)

The availability and consequent increased use of low-cost microprocessor technology has resulted in highly distributed processing systems which most often rely on software support from non-programmer professionals. This trend has resulted in high cost, redundant software development, a lack of consistency, and a multiplicity of problems during operation and maintenance. Therefore, development of appropriate new software development concepts and methodology is required. Tools (in the form of higher level languages and simpler operating systems) which can simplify the software development process are called for to meet the demands which potentially arise from the advent of low-cost microprocessor technology. The need to provide aids for efficient, low-cost software development by non-professional programmers thus represents a Technology Driver of considerable significance.

FAULT-FREE SOFTWARE (PS-03)

Central to the performance of space missions (especially long life autonomous type of missions) is the need for highly reliable software. By definition, fault-free software is software with no internal errors. Of course,

fault-free software alone does not ensure reliable performance but represents only a positive condition for such reliability. External errors in data and hardware can still contribute to the information processing system failures. However, analysis of this facet of the problem, in terms of developing guidelines and methodologies to ensure fault-free software development, reduces the risk of catastrophic failures (such as those which caused, in the early 1970's, a French meteorological satellite to erroneously send destruct signals to 72 out of 141 high altitude weather balloons). Analysis also provides a solid base on which fault-tolerant systems can be built. Hence, fault-free software is deemed a Technology Driver in meeting NASA payload software requirements.

AOL DESIGN METHODOLOGY (PS-04)

The growing use of application-oriented languages (AOL) by a continually expanding user community places a burden on the space-based information processing systems and its management by NASA. This burden, in the form of reduced reliability and higher costs (because of an additional compiler for each new language), is caused principally by a lack of commonality in approach to AOL development. This results in very little consistency and dictates that guidelines and standards for the design of AOL's as well as evaluation criteria for their assessment should be developed. Therefore, NASA should consider this aspect of software engineering as a prime area of concern and accordingly view it as a Technology Driver under this study.

LOW COST DEVELOPMENT OF AOL COMPILERS (PS-05)

This Technology Driver represents an area of concern that is closely connected with PS04 and, in fact, represents a "down the stream" concern from the previous one. Development of AOL compilers and their costs are closely tied to the AOL design aspects in that a better and more consistent AOL design requires simpler and less expensive compiler generation efforts. However, the problem of compiler costs is further complicated by the continued proliferation in the types of computer systems coming into usage. A new compiler is needed not only for each new language but also for each new computer environment in which it is to be deployed. It is therefore necessary to look to this aspect as separate from the problem of AOL design. As such, this is identified here as an independent Technology Driver.

DISTRIBUTED SYSTEM PARTITIONING/INTERCONNECTION TECHNIQUES (PS-06)

Distributed systems, in the context of this description, are defined as systems of multiple processors each having its own executive and memory but performing specific dedicated processing functions as a part of an overall

single partitioned system. While this concept is not new in itself, the progress in hardware technology, particularly the advent of microprocessors, has brought it to the forefront. The advantages of such distributed processing brings in its wake problems of complexity in system design and integration of subsystems. Associated with this facet of information processing are the problems of interprocess communication and optimal distribution of the processing functions among the components. These and other related aspects are covered by this Technology Driver and accordingly viewed as an area needing further attention from NASA in order to meet onboard, real-time, distributed processing needs.

VERY LARGE STORAGE ACCESS SIMPLIFICATION (PS-07)

Availability of very large, low-cost storage, which presently lags behind other hardware technology in its status, can be expected to gain more attention in the coming decade. Accessing very large memories will entail far more complexity than is now necessary. In addition, the number of non-programmers, with the need for such capabilities, will continue to increase, and their needs must be met by appropriate developments in the software area. This need to simplify access to large storage on a routine basis is expected to become critical from NASA's point of view with the increased demand on high-volume, data rate-oriented space systems. Accordingly, it is viewed as a Technology Driver of importance.

SOFTWARE FAULT DETECTION (PS-08)

Software, however much it is checked for faults, cannot be formally proven to be fault-free, especially in the case of large software packages. Further, even the internally fault-free software may fail in operation due to external errors in data and hardware. It is therefore essential that suitable means for fault detection should be inculcated into the system. Such fault detection during execution can alleviate the problem of system failures through initiation of appropriate action to confine the error (from further propagation within the system) and enable recovery. Such recovery and error confinement are very critical to NASA payloads, and as such software fault detection is viewed as a Technology Driver of consequence.

SOFTWARE RECOVERY (AFTER FAULT DETECTION) (PS-09)

This represents the third facet of the software reliability goals to be pursued by NASA to meet its long term requirements. The first facet involves studies aimed at fault-free software, the second is fault detection during execution, and the third is recovery from such faults after detection.

Accordingly, this Technology Driver rounds out those identified earlier as PS-03 and PS-08 and combined the three contribute towards fullfilling the NASA software reliability requirements. Thus because recovery in real time from detected faults is essential to the success of missions, it is viewed as a primary area of concern - a Technology Driver.

HIGH SPEED BUFFERING TECHNIQUES (PS-10)

The earth-oriented nature of future missions will result in onboard data acquisition at very high rates. These projected rates are far beyond the capabilities of present data acquisition, buffering, and processing systems. Consequently, extremely large buffers and associated high speed buffering techniques have to be contemplated in viewing the data systems of the 1980's. High speed buffering techniques are viewed as a Technology Driver of significance to this study.

DESIGN AND CONTROL OF ADAPTIVE SOFTWARE PROCEDURES (PS-11)

Many of the information processing procedures associated with automated intelligence environments such as scene analysis, pattern recognition, and image processing involve a considerably large number of specific processing functions. While each of these may be simple in itself, the many possible configurations of employing them make the task of control of these software procedures complex and challenging. This is the case even if a person is involved in the loop for deciding on the particular configuration most suited to a given data environment. In autonomous environments requiring automated intelligence capabilities, this adaptive selection and execution of specific processing logic modules becomes a critical capability and should be incorporated into the software system. The design and control of such adaptive software procedures is viewed as a Technology Driver of importance to NASA missions requiring autonomous operation capabilities.

NATURAL COMMUNICATION METHODS (PS-12)

Because human resources are at a premium on space vehicles, it is desirable to explore all avenues for minimizing the demand on these resources or, alternatively, expanding the potential of the available resources to the fullest extent. In the case of man-machine interaction, this can be achieved by making the machine capable of interacting with the man in the manner that is most suited to the man, i.e., through natural communication methods. Exploitation of the full potential of the natural communication methods for onboard implementation would result in optimal utilization of human resources. It is therefore desirable for NASA to look into this area, and natural communication methods are thus being identified here as a Technology Driver.

EFFICIENT LARGE ARRAY SEARCH AND SORT PROCEDURES (PS-13)

Most pattern classification software techniques can be efficiently structured by use of table look-up procedures. Such table look-up procedures generally involve large size tables, and consequently both the table formulation and look-up phases involve considerable search/sort operations. Further, these operations are highly data-dependent in terms of the resulting computational loads. The very high data rates associated with the imagery applications and the consequent need for repeated high speed data-dependent table look-up operations dictates that the search/sort operations should be as efficient as possible under the related environment. It is therefore necessary and desirable, from the point of view of enabling onboard image analysis and classification, to evolve efficient large array search/sort procedures.

PARALLEL PROCESSING TECHNIQUES (PS-14)

Parallel processing techniques have come to the forefront with the advent of distributed systems made practical by availability of low-cost microprocessors and related hardware technology. While the general area is under active study, the aspect of exploring its application to image analysis and exploiting its potential in this context has not been given sufficient attention. The concept of parallel processing is especially attractive given the nature of processing involved in the analysis of large images (identical independent processing of a large number of data points in a two-dimensional data array). An in-depth exploitation of the concept in the context of image processing functions is accordingly viewed as a Technology Driver as it leads to more efficient use of the computational resources available onboard and enables onboard implementation of some of these functions (which otherwise would be considered not feasible).

EFFICIENT LARGE ARRAY MANIPULATION PROCEDURES (PS-15)

Data compression, image enhancement, and such other image analysis operations involve two-dimensional processing requiring manipulations such as matrix transposition (of the two-dimensional image data matrix). This problem is important in actual environments involving large images, for it precludes the possibility of holding the total image in core all at once. The desirability of addressing this problem becomes even more apparent when one considers the fact that this manipulation can take significantly more computational effort (and time) than the rest of the processing being carried out on the image. Accordingly, this need for efficient large array manipulation procedures is viewed as a Technology Driver in meeting the NASA requirements of image analysis capabilities.

APPENDIX B

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEETS

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Software Design Engineering — No. PS-01

2. Technology Category: Software Development

3. Missions Affected: All

4. Impact Date: 1978

5. Driver Description: Currently, no facet of software development can be considered as being satisfactorily performed in a production mode, i.e., major shortcomings are identifiable in every step of the development process (with the possible exception of coding). There is a general feeling of inadequacy, dissatisfaction, and a lack of confidence in the merits of proposed solutions. Almost every facet of software development can, therefore, be a Technology Driver. Further investigation into the problems, proposed solutions, and status of technology development reveal, however, a common cause of difficulties. The key to this common cause is the lack of Software Design Engineering technology. Software Design Engineering is conceived to cover interpretation of requirements and creation of a design, but not the physical implementation of the software.

6. Applicability to Payloads: This software design problem is an even more difficult problem in the area of payload software than it is in most other areas. Payload software is generally part of an "embedded" system, that is, a computer system that is only part of, and integrated into, an equipment complex such as an avionics system. The software design problem itself is difficult enough, but in the "embedded" system, these difficulties are multiplied by the external constraints imposed by the total system.

7. Current Technology Efforts: Current and past technology developments have primarily addressed the development cycle from the point that a complete design specification is available. Some effort has been made in formalizing the design specification; however, only recently has recognition been given to the fact that all successes and failures

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

7. (Cont'd) of systems start with the proper design principles or lack thereof. Very little effort has been invested in "embedded" systems. Activity within industry is now centered on development of machine-analyzable, computer-aided requirements analysis software systems (e.g, ISDOS, SREP). These involve problem statement languages and language analyzers. Simulation techniques are slowly being improved. Top Down design is receiving more emphasis. Studies of how to use it and how to represent control structures and data interaction are growing. Modularization and structural design techniques are beginning, although, decomposition rules for system structures are poorly understood and decomposition consistency has not materialized. Better design representations are beginning to appear (e.g., HIPO, PROVAC). Some attempts are being made to develop machine processable design representations (PDL). MSFC is currently developing a Software Specification and Evaluation System (SSES) which shows promise.

8. Current Major Problems: There is a general lack of knowledge of Software Design Engineering as a whole. The design process is not well understood. There is little guidance for designers and little knowledge of what a designer does or how he does it. Software requirements are mostly represented in free-form English specifications which use ambiguous terms and words with unspecified definitions. There is a lack of consistency from requirements to final design specification. Most design is still manual and Bottom-Up. Interface and integration questions are usually not considered until late in the design process if not ignored completely. Top down techniques are slow in developing because so little is known about how to decompose a system structure into proper functions. This is particularly critical with the advent of distributed system techniques in space applications and will continue to be until there are firmly established principles as to how a system should be distributed. Until these problems are resolved, software will continue to be costly and unreliable.

9. Technology Advancements Required: Technology advancements must be made to provide visibility of design, improve design techniques, and establish realistic design evaluation methods. Specifically, technology advances must be made in the following areas:

- Computer-aided design for control structures and data structures.

9. (Cont'd)

- Methods of designing and describing interfaces.
- Development of a better understanding of error mechanisms.
- Development of rules for the decomposition of system structures.
- Development of techniques to enable consistency of decomposition of requirement and specifications descriptions.
- Development of methods of deriving module definitions.
- Improvement to modeling and simulation techniques.
- Development of software prototyping techniques.
- Derivation of design techniques for development of manageable, testable, and maintainable programs.

These areas are of general interest across the industry and are being vigorously pursued; as for example, the development of computer aided requirements and specification language development projects such as the ISDOS project sponsored by the aerospace and government organizations, CARA sponsored by the U. S. Air Force, and SREP sponsored by BMDATC. However, there are some technology items whose developments will have direct bearing on future NASA space activities and to some extent are unique to NASA (in terms of the need to develop real-time distributed systems and distributed data networks for space applications). These are:

- Development of requirements decomposition and structuring guidelines.
- Development of methods of designing and describing interfaces.
- Development of guidelines for design documentation consistency.
- Software prototype methods.

9. (Cont'd) Advances in these areas would provide the knowledge required to partition systems, distribute functions and subsequently determine whether these functions are more appropriate to software or hardware implementations.

The approach basically consists of taking the theoretical work done by L. L. Constantine and others and applying it to the real problem presented by the processing requirements of future space systems.

However, the task is only deceptively simple. At this point in time there are still basic disagreements as to what constitutes proper decomposition of a system. There are basic questions on how to document a design to provide design visibility. But more important, there is no established evaluation criteria to say whether or not a design is good. Establishing these principles is an integral part of the proposed tasks.

These are critical tasks to be addressed by NASA. Failure to do so will foreclose an opportunity to alleviate the continued high cost of software systems of questionable reliability.

Cost of these studies could range from low (3 man years) to medium (10 man years).

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Trend Toward S/W Development by Non- Programmers — No. PS-02

2. Technology Category: Software Development

3. Missions Affected: All

4. Impact Date: 1978

5. Driver Description: With the advent of microprocessor technology in highly distributed processing systems, more and more programming is performed by professional people with no background in programming skills. The programs are preponderantly coded in assembly or even machine language, and are usually unique, single-copy logic (i.e., each is a "new" invention). This trend will continue, and probably at at an accelerated rate, along with its attendant high cost, inefficiency and retraining requirements, unless tools are provided for these systems. The tools required are standard operating systems, high level languages and scientific and mathematical subroutine packages for microcomputers. These tools will allow the individual scientist or engineer to generate reliable software more easily and at a lower cost.

6. Applicability to Payloads: The low cost, volume, weight, and power requirements of microprocessors make them ideal candidates for space systems. These attributes enable new concepts of payload technology to be realized through the distribution and partitioning of functions. Unfortunately, the proliferation of computers expands the need for software and software costs are still rising. The promise of this new technology will not be fully achieved unless ways and means of controlling or limiting software cost can be found.

7. Current Technology Efforts: Available memory on microprocessors has been a limiting factor in converting sophisticated software languages to their use. These limitations are gradually lessening, and industry is beginning to make available higher order languages (primarily FORTRAN) for microprocessor systems. There are also new language techniques, such as FORTH, available to simplify software development. However, very little has been done to standardize and provide off-the-shelf operating systems and software libraries.

8. Current Problems: Microprocessor technology is still in a very evolutionary state (e.g., 4-bit to 8-bit to 16-bit word length). Peripherals are primitive and still expensive. Memory size, though increasing, still imposes limitations on the use of less efficient languages and sophisticated operating systems. In addition, quite often the users of micro (and even mini) systems cannot afford extensive programming staffs. The software now being generated is costly in any case, but much more so when done by non-programmers.

9. Technology Advancements Required: A number of technology advancements are needed to alleviate cost and improve reliability of software created by non-programmers. This is particularly critical to NASA in supporting the rapidly expanding use of space systems. These advancements are possible but need a central, coordinated thrust to achieve fruition. The key to solving current problems is to make programs and programming simpler. The following Technology Items are designed to do that:

- o Development of requirements-to-code translation aids.
- o Implementation of a simplified software development system demonstration model.
- o Research into query guided implementation methods.
- o Classification of error mechanisms.

These are achievable items and achievable at modest cost. Growing support of industry (not just the computer industry) is evident; however, in most cases the efforts at technology advancement is aimed at specific applications which will not satisfy the requirements of the space user community; therefore, NASA should initiate action to pursue these items. Failure to do so will limit the opportunity to slow the continued spiral of software cost and enhance software reliability.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Fault Free Software — No. PS-03

2. Technology Category: Software Development

3. Missions Affected: All

4. Impact Date: 1978

5. Driver Description: Software reliability (or unreliability) is a subject of universal concern, but it is an overriding concern for space flight particularly on longlife, autonomous missions. Fault free software is software that is correct and its internal data contains no errors. This does not necessarily equate to system reliability for the software is still subject to external data and machine errors; however, it definitely removes software per se as a cause of system unreliabil-

6. Applicability to Payloads: Dependable systems are mandatory for Space applications and systems are never more dependable than their weakest component part. Software is a unique component of these systems because it is so difficult to predict where or when it might fail; conversely, if it could be completely "proven," it would never fail. This is what NASA should strive for.

7. Current Technology Efforts: As software reliability is of universal concern, the search for fault free software is a universal activity. Many avenues are being explored in the development of new and improved testing techniques, better requirements and design approaches, mathematical theory and management practives. Even computer architecture is evolving which aids in the prevention of software errors, but with all this activity truly fault-free softw is hard to find.

8. Current Major Problems: The basic problem to be solved is how to develop adequate Software Engineering Techniques. Of course, other problems exist in areas such as validation/verification, lack of requirements and lack of standardization, but these will probably never be solved until Software Engineering comes of age.

9. Technology Advancements Required: First and foremost, the development and dispersion of Software Engineering Technology. This should be accompanied by advances in other areas:

- o Consistency testing techniques to prove that design reflects requirements.
- o Automatic test case generators.
- o Integration and traceability techniques.
- o Study of error mechanisms.
- o Static and dynamic code checks in higher order languages.
- o Program proving aids and theory.
- o Improved quality assurance techniques.
- o Software redundancy techniques.
- o Computer architecture refinements.

Most of these areas are being addressed both within NASA and outside, as for example SSES of NASA-MSFC. However, there is one additional area of significance that only NASA can address. This item is standardization. Standardization has many implications. It implies standard languages, standard operating systems, standard interface implementations and consistent management practices. Not all of NASA software can be standardized, but that, which can, should be identified and placed under rigid configuration control.

One characteristic that software evinces is the fact that the probability of a failure happening is proportional to the number of errors that exist in a program. Consequently, as failures occur and errors are removed over a span of time, the more reliable the software becomes, until theoretically it becomes fault-free. This is one of the unique and valuable assets provided by standardization. The use of standard functions multiplies the opportunity to exercise the code and allows it to be exercised in many different environments. Though a standard function may not initially be fault-free, given time it will certainly approach a fault-free state.

Standard functions for aerospace applications can be identified and developed. There is virtually no support of standard aerospace software functions within the industry. Therefore, one Software Technology Item that NASA should address is:

- Development of standardization criteria for NASA software.

Cost will vary with the degree of standardization, but total future software development cost would be significantly reduced. This area will also be aided indirectly by the technology item - classification of error mechanisms identified previously under PS-02.

Failure to standardize will be expensive and prolong the risk of catastrophic failure due to software error.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: AOL Design Methodology No. PS-04

2. Technology Category: Software Development

3. Missions Affected: All

4. Impact Date: 1978

5. Driver Description: A shift to an increased emphasis on application-oriented languages must be anticipated. This is due to the growing number of engineers (as opposed to professional programmers) developing or writing programs. Language design is one of those typical software areas which have no real guidelines, standards, or evaluation criteria which can aid in the quick development of high order languages. Language design is frequently a costly process of long duration; therefore, the development of an application-oriented language design methodology must be considered as a NASA goal.

6. Applicability to Payloads: The broad dispersion of the space vehicle user community, coupled with the increasing number of programs being written by engineering personnel rather than professional programmers, will demand the development of application-oriented languages for writing space application software.

7. Current Technology Efforts: Virtually no technology efforts exist with the exception of work with extensible languages and compiler writing systems.

8. Current Major Problems: Compiler writing systems have not fully matured. Extensible languages, although a subject of interest for years, still have severe limitations and will probably never fulfill expectations. Current language design methodology is expensive. There is a need to discover what AOL requirements are and to discover how to get a good design.

9. Technology Advancement Required:

- Definition of guidelines, standards and evaluation criteria for the development of higher order languages.
- Development of concepts for the development of requirements of an AOL.
- Development of language design concepts.
- Development of problem language statement and analysis techniques.
- Evaluation of adaptive HOL's and extensible languages.
- Development of simple languages.

Effort expended on these items will result in evolutionary advancement. It is doubtful that any major breakthroughs will be achieved. Nevertheless, whatever gains made would be of benefit to the user community and would tend to reduce costs and add to reliability.

One Software Technology Item that NASA should address is:

- Development of AOL design guidelines/standards.

Failure to expend this effort will impose higher than necessary software costs on users of space systems. Guidelines for Design Structure consistency identified under PS-01 as a technology item can also be expected to contribute towards AOL design methodology at least in an indirect sense.

1. Technology Driver: Low Cost Development of AOL Compilers — No. PS-05

2. Technology Category: Software Development

3. Missions Affected: All

4. Impact Date: 1978

5. Driver Description: For each language and computer environment pair, a specially tailored compiler is necessary. Although considerable research and development effort has been expended, such compilers have remained as very expensive items. The potential proliferation of languages and computer systems obviously requires that these costs be decreased significantly. This Technology Driver is closely related to Technology Driver PS-04, but here the emphasis is on development cost reduction as opposed to design methodology.

6. Applicability to Payloads: A convenient, low-cost method of generating applications-oriented languages is required to support the space system user community. This is particularly so for users who have limited or minimal resources to pursue space exploration. In addition, in terms of software reliability, there is a need to ensure consistency in the way in which AOL's are implemented.

7. Current Technology Efforts: Primarily, these are limited to government-sponsored efforts to produce compiler writing systems, metacompilers and language translators. Some activity on extensible languages (mostly by Universities) and industrial application of new language technology (e.g., FORTH by FORTH, Inc.), has also been reported.

8. Current Major Problems: Compiler writing systems have not fully matured. Language translators have been found to be difficult and expensive to build and are not as universal as had been hoped. Extensible languages have not fulfilled expectations and probably never will except for specific applications.

9. Technology Advancement Required:

- o Advancements in intermediate language technology.
- o Identification of the principles for using directory techniques for development of AOL's.
- o Improvements to the technology of adaptive HOL's and extensivle languages.
- o Continued development of compiler writing systems and language translators.

Evolutionary improvement of these items will reduce cost of AOL's if the improvements are aimed at cost reduction. In general, though, the key items are to determine what the application-oriented language requirements really are and find out why compiler generators cost so much. The most critical item for NASA at this time (in addition to the one identified previously under PS-04) is:

- o Evaluation of AOL compiler generator cost factors.

Failure to expend this effort will place added burden on the user community and perhaps close the door to those with limited resources.

Cost of pursuing this effort would be moderate (3 man years).

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: (Distributed) System Partitioning/ Interconnection Techniques No. PS-06

2. Technology Category: Software Systems Architecture

3. Missions Affected: All

4. Impact Date: 1978

5. Driver Description: A distributed system, as the term is used here, is a system which contains multiple processors, each with its own executive, performing dedicated functions as part of a single partitioned system, usually each housing its own main memory. The concept is not particularly novel. However, it is only recently, through the advent of microcomputers, that it has become eminently practical.

A distributed system has two major advantages:

- o Simplified development/modification of the system.
- o Ability to perform parallel processing.

Simplified development/modification of the system comes about through the relative independency of the subsystems. That is, the computer process becomes a more integral part of the subsystem to which it is assigned and thus it can be designed, adapted, and verified almost independent of the other subsystems; however, the system design as a whole does not become any simpler, only more flexible. The very flexibility may make the system design, or the choice of system design, more difficult. Any network of computers has, for the software as well as the hardware, the inherent problem of selecting interprocess communication techniques and selection of functions to be or not to be distributed. This is, generally, an ill-defined area and thus results in a Technology Driver.

6. Applicability to Payloads: Virtually all future payloads will use distributed systems in one form or another, if they have computer systems at all. In addition, this concept will be the enabling function of intelligent instruments. System partitioning and interconnection techniques are particularly crucial in development of real-time systems.

7. Current Technology Efforts: Distributed systems are not new and the rapid growth of this technology, made possible by the availability of low cost processors, has stimulated a great deal of research and study both within and without the industry. However, a disproportionate amount of this study has been directed at what the technology can be applied to as opposed to how to apply it. The Unified Data System research supported by NASA is a good example of the current technology efforts.

8. Current Problems: Selecting interprocess communication techniques, identifying functions, and selecting those functions to be distributed present the major problems. However, development of user-oriented software development tools and methods must be faced to bring the cost of distributed system software down to an appropriate level. There is a critical need to develop rules and methods for decomposing systems for optimum and consistent partitioning.

9. Technology Advancements Required:

- o Research into optimal task allocation and scheduing methods.
- o Development of control structures for implementing these schedules.
- o Development of techniuqes for concurrent processes.
- o Development of NASA-oriented functional hardware/software interface technology.

There are two items arising from these areas which must be accomplished by NASA. These are:

- o Development of interface criteria for NASA distributed processor applications.
- o Development of task control structures for distributed environments.

Cost of these efforts will be modest (3-6 man years each). Failure to pursue these items will severely jeopardize the potential inherent in distributed system applications. This facet of Software Engineering is also furthered by decomposition and structuring guidelines identified earlier as a technology item under PS-01.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Very Large Storage Access Simplification — No. PS-07

2. Technology Category: Software System Architecture

3. Missions Affected: Imaging Missions, Autonomous Missions

4. Impact Date: 1985

5. Driver Description: A hardware technology that is associated with the availability of low-cost computers, but currently lagging is the availability of low-cost, large, rapid-access storage. Accessing such memories and using them effectively is currently a specialized software area. To enable non-programmers to use such storage devices on a routine basis requires some method of simplifying their use.

6. Applicability to Payloads: With the advent of a data explosion, particularly in Earth and ocean observation, the need for image preprocessing and onboard classification is very real. In addition, deep space, autonomous payloads will demand high level data compression onboard. These processing functions require very large storage capability for the manipulation of large single and multidimensional arrays, intermediate storage (buffering) and reference data.

7. Current Technology Efforts: Research in data base technology is very expensive and theoretical research is beginning to decrease. However, within the industry, a considerable effort is being invested on the development of improved access methods, more efficient control and recovery techniques and data structures. The most recent data base technology has been directed toward the development of relational data bases as opposed to hierarchical or network approaches. These efforts are directed at large ground systems and do not particularly apply to the class of use envisioned for space applications. There is literally no research or study directed toward the application of very large storage systems for onboard applications, nor is there any study as to how these systems would be used even if they were available.

8. Current Major Problems: Although data utility constitutes a vitally important area for the future space applications, it presents problems so complex that they are even hard to state. Very little data exists

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

No. PS-07 Sheet 2 of 2

8. (Cont'd) on the definition and use of these systems. Little is known about how the data structures are defined, how they are to be accessed, and precisely how they will be used. There has been very little research into large storage devices for real-time onboard systems.

9. Technology Advancements Required: There are many areas in which technology development should begin. The more important of these are:

- o Develop techniques for representing data structures in computer-aided software design.
- o Develop structure techniques that will support high rate processing.
- o Develop techniques for data base sharing in distributed systems.
- o Develop concepts for parallel access methods.
- o Develop concepts to enhance data independence.
- o Evaluate the use of data storage preprocessors.
- o Identify error confinement and recovery techniques.

These items are important; however, except for research into real-time onboard systems (which is a hardware gated technology) it is felt that at this time all these areas are adequately addressed. NASA should monitor the technology advancements in these areas so as to be able to apply this technology when hardware for space systems becomes available. Accordingly, this monitoring becomes the main technology item here. However, this item is common to other areas across the board as shown in the technology driver-item correlation table.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Software Fault Detection No. PS-08

2. Technology Category:

3. Missions Affected: All

4. Impact Date: 1980

5. Driver Description: It may, for various reasons, never be truly feasible to provide absolute proof of software correctness. It therefore behooves us to study an alternate means of preventing the software from causing a system failure. This approach is detection of software failures during execution. It is not clear if and how this is truly feasible; nevertheless, it is the only currently known alternative and must therefore be listed as a Technology Driver. Fault detection is critical for two major reasons:

   - o Confinement of damage, and
   - o Enablement of recovery.

   The more critical a function the greater is the need for fault detection capability.

6. Applicability to Payloads: Real-time applications are particularly demanding of software reliability. The output must be correct the first time and on time. Typical of these applications is guidance and control of spacecraft, and in some cases the collection, processing, and distribution of data. Software failures can cause immediate, significant losses and have been known to do so in the past. In any event, even if failure cannot be prevented, early detection can help to minimize losses and in some cases pave the way to error correction. This certainly has implications for future payloads.

7. Current Technology Efforts: To date, most failure detection technology has concentrated on hardware methods such as microdiagnostics, on-line diagnostics, parity checkers, comparaters and watchdog timers. The only real software error detectors have fallen into the category of error detection/correction codes, sum-checks and reasonableness tests. Just recently, industry has begun theoretical research into error mechanisms, software redundancy techniques and sophisticated error logic within

7. (Cont'd) software structures. There has been virtually no attempt to use these concepts in practical applications.

8. Current Problems: It is difficult if not impossible to suspend processing at the point at which a failure occurs. Usually several instructions will be executed after a failure and the results are unpredictable. There is no guaranty that an error will not also invalidate proper operation of the error detection software. Redundant software is costly both in life cycle cost and in system overhead.

9. Technology Advancements Required: It is difficult to see specific technology advancements to improve software fault detection methods; however, a better understanding of error mechanisms should lead to better predictability of how and where errors are likely to occur. The most significant gains will probably come through improved computer architecture. Isolation of processes (modules, subfunctions, etc.) must be improved to at least limit the damage when an error does occur. In general, following good programming practices - knowing and understanding the relationship of inputs to outputs - will do more for error detection than perhaps any other facet of software technology. Some items currently under consideration by industry are:

- o Fault criteria determination.
- o Reasonableness testing concepts.
- o Software redundancy voting techniques.
- o Analysis of error mechanisms.

Of these, the one item (already identified as a Software Technology Item under PS-02) of major significance here is:

- o Research into the classification of error mechanisms. (Items identified under PS-01 are also applicable to solution of the error detection problem.)

9. (Cont'd) The cost of this research will be moderate (3-6 man years). Documented errors from previous space programs should be used to consolidate these errors as a whole in order to throw new light on where errors occur and why they occurred.

Failure to pursue this study will limit the application of industrial developed technology advancements to NASA space systems.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Software Recovery (After Fault Detection) No. PS-09

2. Technology Category: Software Systems Architecture

3. Missions Affected: All

4. Impact Date: 1980

5. Driver Description: Associated with the problem of fault detection is the problem of fault correction and recovery. In many hardware schemes, and in all known software schemes of fault detection, one or more instructions have been executed by the time a fault is detected. Recovery from such unwanted instruction execution is still an unsolved problem and must therefore still be considered a Technology Driver.

6. Applicability to Payloads: Software recovery techniques after error detection are very much applicable to payload software particularly in view of the fact that most of this software falls in the category of real-time systems. The ability to recover from error could make the difference between success or failure of a space mission.

7. Current Technology Efforts: Active research in software recovery techniques basically consists of attempts to improve older technologies such as checkpoint/restart or to impose proven hardware reliability concepts (like redundancy) on software. The newer concepts include a technique using recovery blocks which contain needed variable data and alternate processing algorithms, reconfiguration and retry (made possible by new low cost hardware), and system designs that provide small tightly closed environments. All of these approaches add considerable system overhead.

8. Current Major Problems: It is very difficult, if not impossible, to determine what happens between the time an error occurs and the time it is detected. This prevents acquiring a precise knowledge as to how far the process must back track (or to what hierarchical level) in order to fully recover. Overhead is too high to support most real time functions without excessive loss of data on error occurance.

C-2

9. Technology Advancements Required: A number of new technology advancements are required for this relatively undeveloped field. These include:

- o Control structures for dynamic software reconfiguration.
- o Development of software techniques to back up or mask hardware failure.
- o Development of techniques for restoring data integrity.
- o Development of techniques for identification of easily recoverable checkpoints.

These areas are getting more and more emphasis not only within NASA but also in industrial and university research; however, NASA should concentrate on an item of key importance to space applications. This is:

- o Develop program organization methods for real-time fault recovery.

Cost of this research will be at a moderate level although extended in time (3-6 people for 3-5 years).

Lack of effective recovery in real time space activity constantly places missions in jeopardy or requires sophisticated and expensive redundant hardware systems. The cause of recovery techniques will be advanced to some extent by the classification of error mechanisms identified in PS-02 and the structuring guidelines identified in PS-01.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: High Speed Buffering Techniques No. PS-10

2. Technology Category: Software System Architecture

3. Missions Affected: High Data Rate Imaging Missions

4. Impact Date: 1983

5. Technology Driver Description: The very high data rates expected in the future missions (involving acquisition of multispectral image data over vast areas) which are beyond the capability of present day onboard computational resources, calls for large buffers and consequently high speed buffering techniques is viewed as a technology driver in meeting the requirements of data acquisition and processing on such missions.

6. Applicability to Payloads: This is mainly applicable to missions involving high data rate imaging experiments.

7. Current Technology Efforts: Buffering techniques are currently under study from various viewpoints, most important of which is that of its role in a communication based computer system. These studies include methods of core allocation and associated problems in queuing theory, etc. Related efforts in hardware technology areas in terms of advancements in storage techniques are also under progress.

8. Current Major Problems: The needs of the high volume data experiments of imagery acquisition and analysis cannot be easily met with the current levels of buffering capacities in terms of size and speed and as such high speed buffering techniques tailored to the qualitative and quantitative aspects of onboard image analysis experiments are called for here.

9. Technology Advancements Required: The efforts required to meet these needs can be summarized as studies aimed towards the development of:

   - Adaptive high speed buffering techniques for deployment in dynamic networks.

   Included within the scope of these studies will be topics such as store

and forward techniques, data aging, and other related aspects (such as change detection) in the context of image acquisition and analysis, and processor architecture and peripherals characteristics (such as use of natural communication networks) effects on high speed buffering techniques.

These efforts, expected to be in the neighborhood of 3-5 man years, are considered essential for achieving onboard image acquisition and analysis capabilities at the data rates projected for future missions by the OFS Study. Supportive efforts include monitoring of development in allied fields such as processor architecture and advanced storage techniques which are likely to influence the proposed technology efforts.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Design and Control of Adaptive Software Procedures — No. PS-11

2. Technology Category: Software System Architecture

3. Missions Affected: Advanced missions involving some level of automated intelligence activities

4. Impact Date: 1988

5. Driver Description: The software procedures associated with automated intelligence, scene analysis, image processing and pattern recognition activities are numerous in number and complex in their interactions. Depending on the specific problem environment, these need to be put together in different combinations and configurations. The configurations are determined by the human in the loop on the basis of intermediate results during the processing. However, in automated intelligence environments, this adaptive selection and implementation of specific processing logic modules has to be incorporated into the software system as an integral part of the system. The design and control of adaptive software procedures is hence a driver for accomplishing these automated intelligence objectives.

6. Applicability to Payloads: Mainly, this affects payloads expected to operate in deep space unmanned missions in ill-defined environments.

7. Current Technology Efforts: Most software technology efforts in the areas of automated intelligence, image processing and scene analysis, have concentrated on the development of specific software to perform well defined individual tasks with a human in the loop dictating the adaptive selection of these specific software in the dynamic environment. Few reported efforts are available on the problem of integration of these software procedures into a system with adaptive capabilities.

8. Current Major Problems: The areas of automated intelligence itself being in an embroynic stage, the system concepts are not yet fully developed and the requirements are not well defined. The problem is therefore to develop a fuller understanding of the objectives and requirements of the software system needed to support such activity followed by efforts at developing the means of meeting these requirements.

9. Technology Advancements Required: Much of the efforts needed in this area being still at conceptual levels, the initial effort should be:

   - Definition and development of the control structure concepts for adaptive software procedures.

   Other efforts which can advance the cause of this study are development of techniques for adaptive assembly of microprocedures and dynamic path strategy selection as required in PS-10 and PS-13.

   These advancements are deemed to be basic to the successful operation of deep space unmanned exploratory missions planned for the next decade and are likely to involve 3-5 man years. Supporting activities include monitoring of the developments in the overall area of automated intelligence.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Natural Communication Methods No. PS-12

2. Technology Category: Software Applications/System Architecture

3. Missions Affected: All Manned Missions

4. Impact Date: 1980

5. Driver Description: In view of the naturally severe constraints on the availability of human resources in space environments, it is essential to make full use of the potential of the available resources. It is therefore advisable to provide for all possible natural communication methods to enhance the capabilities of human interaction with the computer system for all command and control type of activities. This enables maximum utilization of all the faculties at the command of the human operator. Exploration of the scope of natural communication methods in space environments is therefore a technology driver in achieving maximum utilization of man-machine systems.

6. Applicability to Payloads: This is mainly applicable to those involving human monitoring or performing of multiple operations which thereby place a heavy demand on human resources.

7. Current Technology Efforts: Use of natural communication methods between man and machine is indeed a popular area of industrial investigations. However, the needs of space applications as well as the associated constraints are not the same. Detailed studies of this aspect except perhaps for specific mission requirements, have not been reported. Further, the use of the natural communication methods is bound to affect the payload software system characteristics and this needs to be studied in depth.

8. Current Major Problems: The first problem is that no comprehensive study of the needs for natural communication methods in space environments is available. Such a study if carried out can identify the specific types of communication methods of value in space environments. Use of such natural communication methods will have to be integrated within the payload software system and this represents the major software related problem of concern here.

9. Technology Advancements Required: The needs of this technology driver is best served by a preliminary study to identify the specific natural communication methods of significance in space environments. This can be accomplished by a review of both the needs for natural communication methods in space environments and the spectrum of natural communication methods known to be successful in other environments. This is then to be followed by a software technology related study whose objective will be:

    - Impacts of natural communication methods on NASA payload systems.

This is viewed as an indispensible development for the success of future complex manned missions involving ordinary civilian personnel and is expected to involve about 2 to 3 man years of effort. In addition, this effort should be supported by a monitoring activity of the developments in man-machine natural communication methods such as voice communication (recognition) techniques and containment of storage density for voice patterns.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Efficient Large Array Search and Sort Procedures — No. PS-13

2. Technology Category: Software Applications/System Architecture

3. Missions Affected: All those involving pattern classification of multi-spectral image.

4. Impact Date: 1985

5. Driver Description: Many pattern classification software techniques involve table look up procedures with large arrays (tables) for efficient implementation. This, in turn, involves use of search and sort procedures in the table formulation and look up phases. In view of the expected high data rates in image applications and consequent need for high speed (repeated) table look up operations, it is desirable to achieve maximum efficiency in these operations. This will help in gaining considerable savings in computer resources requirements and will facilitate real time onboard implementation of these classification software. Accordingly the need for efficient search and sort procedures is considered a technology driver. Further, this will also be of concern in NASA software standardization.

6. Applicability to Payloads: This is of particular significance in all payloads wherein pattern classification (of large volumes of multi-spectral data) is called for by the experimental objectives.

7. Current Technology Efforts: Very many search and sort procedures have been reported in the literature mainly in the context of their application in operating system software. However, the needs and constraints as well as the array data characteristics are not necessarily the same in these applications. Very little effort in tailoring or restructuring these methods to the needs of pattern classification problems has been reported.

8. Current Major Problems: The major problem is the need for computationally more efficient search and sort software procedures specifically designed to meet the requirements of the pattern classification algorithms environment, which in addition, are adaptive to array data characteristics.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

9. Technology Advancements Required: The major effort required can be stated as:

   o Development of adaptive search and sort procedures.

This can be accomplished by a planned organization of the following activities: (1) review of all reported search and sort procedures to assess their relevance and applicability to pattern classification algorithms environment, (2) selection and assessment of candidate techniques in the context of pattern classification at the expected data rates of payload experiments, (3) study of the interrelationships of search and sort procedures with data storage/retrieval techniques and processor architecture, (4) study of the feasibility of deriving adaptive procedures to make efficient use of the array data characteristics, (5) using the results of these different supporting studies, develop adaptive search and sort software techniques which are best suited to specific space system configurations and problem environments.

These developments, which can be expected to involve 2 to 3 man years of effort, are viewed as a prerequisite to onboard implementation of pattern classification and such other related software techniques which are needed for multispectral image data analysis experiments. In addition this will contribute towards NASA software standardization. As a supporting activity, NASA should monitor the development in pattern classification techniques particuarly the table look up approaches to ensure that the proposed technology efforts are in step with other developments in the area.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

1. Technology Driver: Parallel Processing Techniques No. PS-14

2. Technology Category: Software Applications/Software Architecture

3. Missions Affected: All involving image analysis.

4. Impact Date: 1983

5. Driver Description: Many of the image processing, pattern recognition and scene analysis functions in the context of image data analysis and automated intelligence activities can be visualized as a set of processing operations carried out independently on the different pixels in the image. Accordingly, these operations can be viewed as parallel processable instead of in sequence as is generally the case. While the concept of parallel processing is in itself not new, an indepth exploitation of this concept in the context of image processing functions is viewed as the technology driver, for achieving efficient use of computational resources onboard and facilitating onboard implementation of some of these functions which otherwise would be infeasible. This will also help in achieving standardization of software of concern to NASA.

6. Applicability to Payloads: This is mainly applicable to the advanced mission payloads involving experiments of image acquisition and analysis.

7. Current Technology Efforts: Parallel processing techniques are indeed a fairly active area of study. Most efforts are however in the general field of parallel processing and associated conceptual and hardware development. Very little of this effort is directed toward exploitation of this concept in implementing image processing techniques. While parallel processing implementations of particular algorithms have been attempted, no general study, which comprehensively explores the scope of parallel processing in the context of image data analysis, is known.

8. Current Major Problems: The major problem is thus the need to apply available parallel processing concepts with necessary and desirable modifications to the entire spectrum of image processing algorithms to achieve maximum utilization of these concepts and thereby meet the severe demands of onboard image processing.

9. Technology Advancements Required: The needs of this technology driver can be satisfied by:

   - Restructuring major software techniques (of relevance to NASA image processing and pattern recognition activities) for parallel processing.

This is accomplished by a prior review of major image processing and pattern recognition functions and algorithms to identify those which are "parallel processable" directly or through modifications.

Another associated effort of significance here (discussed earlier in PS-06) is that of optimal task allocation and scheduling and associated control structure development in the context of parallel processing environment.

Other studies which would be beneficial include fault detection and recovery in parallel processing environments (through reconfiguration of parallel processor modules) and addressing problems of software synchronization. Accordingly progress in these areas have to be monitored by NASA.

These studies are necessary for accomplishing image acquisition and analysis onboard as required by the future imaging missions, and are likely to need 3-5 man years of effort.

SOFTWARE TECHNOLOGY ITEM DERIVATION WORKSHEET

Efficient Large Array Manipulation

1. Technology Driver: Procedures No. PS-15

2. Technology Category: Software Applications/System Architecture

3. Missions Affected: All Involving Image Analysis

4. Impact Date: 1983

5. Driver Description: Data Compression, Image processing and scene analysis functions often involve manipulation of large two (and sometimes three) dimensional data arrays, for example transposition of a large image matrix. This problem is non-trivial whenever the size of this array is too large to be held in core all at once. This obviously is the case when, say, we are processing an (1024 X 1024) image. Manipulations of such large arrays, being an integral part of many data compression and image processing procedures, is viewed accordingly as a technology driver.

6. Applicability to Payloads: In view of the specialized nature of the problem, this affects mainly those payload missions which involve such image acquisition and analysis programs.

7. Current Technology Efforts: There have been a few studies directly related to this problem of matrix transposition. Various alternatives have been proposed, most of which aim at reducing the number of transputs (inputs and outputs) required for transposing the matrix available externally in a sequential or random access storage device. However, the claims and counterclaims of these alternatives have not been evaluated. Another aspect which has been studied to some extent is that of reducing the size of the additional storage required for performing such transposition. There also exists a study aimed at investigating the problem of the so called optimal partitioning of the matrix to facilitate its manipulations. All of these efforts are only directed towards one or the other aspect of the problem in the context of large processing systems available on ground.

8. Current Major Problems: The major problem is to make the manipulations of large two (or three) dimensional arrays practical on board, under the related constraints of time and computational resources such as core and external memory, for facilitating data compression, image processing and scene analysis activities.

9. Technology Advancements Required: Technology efforts are needed to develop a well defined procedure for optimal partitioning of the higher dimensional arrays, which will not only take into account (onboard) data processing system configurations and constraints, but also exploit to the full extent the array data characteristics. This can be achieved by a study of these different aspects: (1) feasibility of development of optimal partitioning procedures for two and three dimensional arrays under specified system constraints. (2) adaptability of array manipulation procedures to array characteristics. (3) interrelationships of array manipulation procedures with data storage techniques and processor architecture.

These efforts (which will approximate to 2-3 man years) can be summarized as a study for the development of

- Optimal large array partitioning procedures.

This is very essential for onboard implementation of most data compression and other image processing techniques of importance to multispectral image data analysis. In addition, the progress in the area of electronic cyclic memories (which has a definite bearing on problems like matrix transposition) needs to be monitored by NASA to gain full advantage of the progress in hardware technology.

APPENDIX C

Cost/Time/Priority Assessment Worksheets

This appendix presents the Cost/Time/Priority assessment carried out for each of the technology items derived from the technology drivers under Phase II of the Study. Each item is described in terms of its objectives and goals, alternative solutions and related studies are identified, and NASA efforts, schedules, resources and priorities are assessed. These details are presented in the form of worksheets in this appendix.

COST/TIME/PRIORITY ASSESSMENT

TITLE: Requirements Decomposition and Structuring Guidelines NO. TI-01

1. DESCRIPTION OF ITEM: Development of proper and consistent requirements decomposition and structuring guidelines for the design of real-time distributed systems for space applications. A major problem in the decomposition and structuring of software design is the lack of consistency from designer to designer. This makes it very difficult to identify proper design and almost impossible to evaluate software systems in any way, except in an empirical fashion after the software has been developed. This point in the cycle is too late. Errors detected after development are orders of magnitude more expensive to correct. This Technology Item seeks methods to eliminate this problem. Evaluation criteria for the designs and design consistency will have to be developed.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 3-4
3. REQUIRED LEVEL: 6 DATE 1980
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

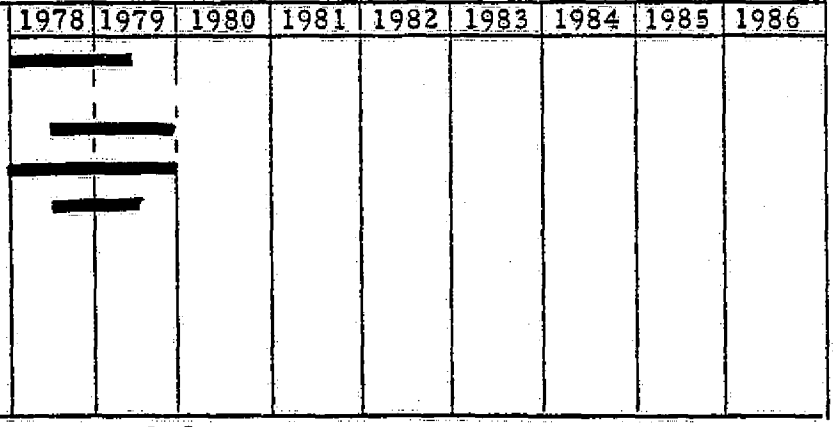
5. ALTERNATIVE SOLUTIONS:
   (1) Development of requirements-to-code technology, thereby bypassing the design phase.
   (2) Utilization of automated design tools.

6. RELATED ACTIVITIES:
   TI-08 Interface criteria for NASA distributed processor applications.
   TI-09 Task control structures for distributed processor environments.
   TI-18 Guidelines for design documentation consistency.
   TI-19 Software prototyping methods.

7. RECOMMENDED NASA EFFORT: Initiate a study effort by calendar year 1978 to establish guidelines for decomposition methods. Couple this with development of evaluation criteria, structuring guidelines and design consistency guidelines to support activities from 1980 - 1990.

ESTIMATED DEVELOPMENT COSTS: $150K - $500K PRIORITY: 5

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Requirement-To-Code Translation Aids NO. TI-02

1. DESCRIPTION OF ITEM: Automated aids (such as RPG) are to be developed for translating well defined payload software requirements into program code (in the desired language). In effect, the translation aids may be looked upon as a set of master programs which output specific program codes corresponding to a given input of software requirements.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2
3. REQUIRED LEVEL: 6 DATE 1981
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 3-4

5. ALTERNATIVE SOLUTIONS:
   - o Continued use of the more expensive, error prone manual approach to translating into code the specified software requirements, is the only alternative available at the present time.
   - o Query guided methods is another alternative being proposed as a parallel task (TI-04).

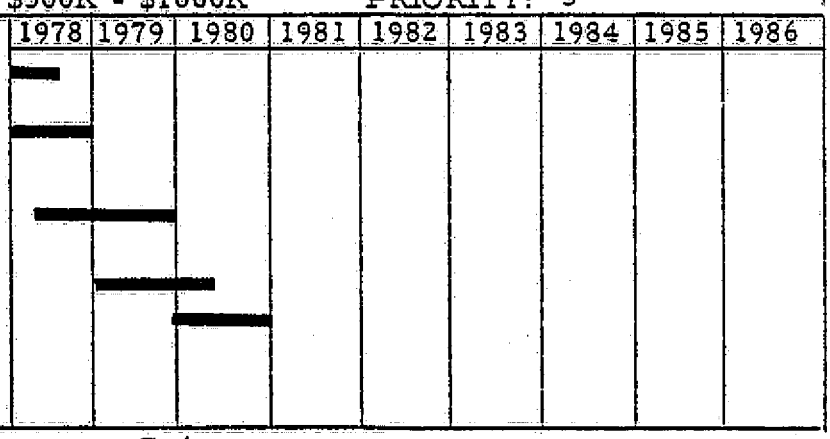
6. RELATED ACTIVITIES:

   TI-01 Requirements decomposition and structuring guidelines.
   TI-03 Simplified software development system demonstration model.
   TI-04 Query guided implementation methods.
   TI-19 Software prototyping methods.

7. RECOMMENDED NASA EFFORT: NASA efforts should be to initiate a study starting in 1978 to investigate the feasibility of development of these automated aids to meet NASA payload software requirements and follow it up with the development of such aids along the lines identified as feasible in the first stage of the study. The detailed schedule of this development is shown below.

ESTIMATED DEVELOPMENT COSTS: $500K - $1000K PRIORITY: 3

8. SCHEDULE:
   - o State-of-the-art assessment
   - o Payload software requirements identification
   - o Master software instructions library development
   - o Transformational approach development
   - o Test and documentation



COST/TIME/PRIORITY ASSESSMENT

TITLE: Simplified Software Development System Demonstration Model NO. TI-03

1. DESCRIPTION OF ITEM: Creation of a demonstration model of a simplified software development system. This item is aimed at lowering the cost of software for users of microprocessors in distributed systems for real-time space operations. The demonstration model will be a precursor of low cost operational systems for non-programmer PI's.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2
3. REQUIRED LEVEL: 6 DATE 1981
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 3

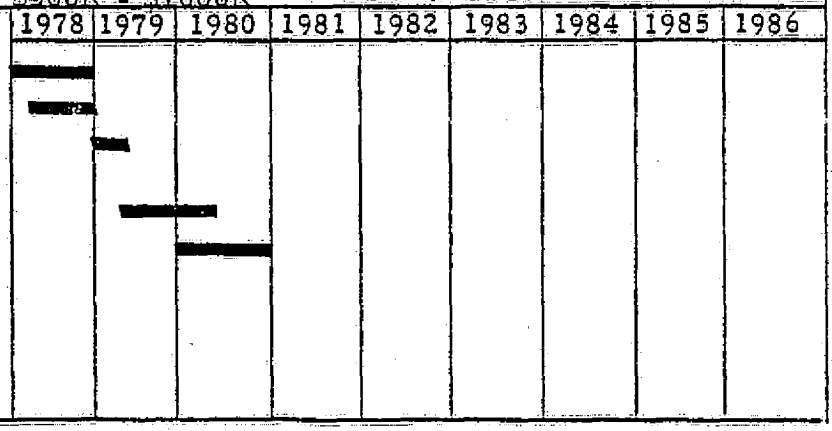
5. ALTERNATIVE SOLUTIONS:
   (1) Query-guided implementation methods.
   (2) Requirements-to-code translation aids.

6. RELATED ACTIVITIES:
   TI-02 Requirements-to-code translation aids.
   TI-04 Query-guided implementation methods.
   TI-19 Software prototyping methods.

7. RECOMMENDED NASA EFFORT: Begin requirements definition for a simplified software development system demonstration model in calendar year 1978. Schedule analysis, design, development and test such that a full year of evaluation can be accomplished in 1980. This will allow 2 years (1981-1982) for development of an operational system to support heavy space activity 1983 - 1990.

ESTIMATED DEVELOPMENT COSTS: $500K - $1000K PRIORITY: 3

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Query Guided Implementation Methods NO. TI-04

1. DESCRIPTION OF ITEM: New programming procedures have to be developed based on the idea of using a questionnaire approach to the programming task. The questionnaire approach helps in guiding the programmer through the different steps involved in his effort and thereby eliminates the errors of omission and reduces the errors of commission. An example of questionaire approach would be the PSI Program Synthesis System (of Stanford University). Such an approach should be adapted to the needs of NASA payload software technology.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2-3
3. REQUIRED LEVEL: 6 DATE 1983
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

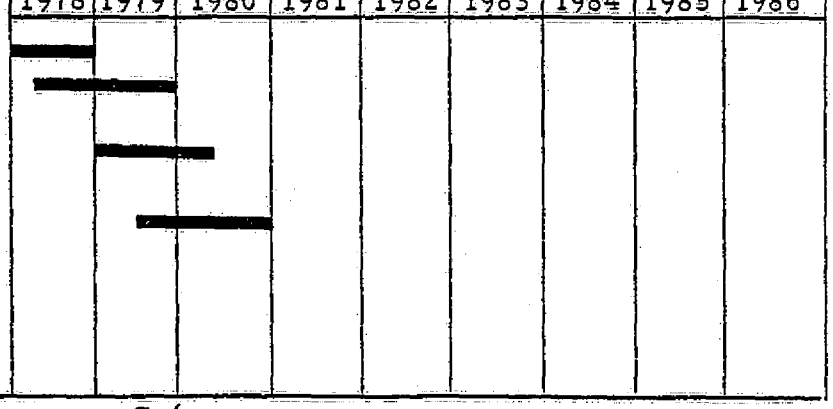
5. ALTERNATIVE SOLUTIONS:
   - o Use of the present day unguided manual approach involving considerable scope for both errors of omission and commission is the only available alternative.
   - o The proposed Requirements-to-Code translation aid is a possible alternative but has yet to be practically demonstrated.

6. RELATED ACTIVITIES:
   - TI-02 Requirements-to-Code translation aids.
   - TI-03 Simplified software development system demonstration model.
   - TI-19 Software prototyping methods.

7. RECOMMENDED NASA EFFORT: The NASA effort required here spans the spectrum of NASA payload related software activities and is an R&D study to be initiated in 1978. The activity proposed here parallels the one proposed under TI-02 to some extent.

ESTIMATED DEVELOPMENT COSTS: $200K - $500K PRIORITY: 3

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Standardization Criteria for NASA Software NO. TI-05

1. DESCRIPTION OF ITEM: Standardization of NASA software is required to eliminate redundant development with its attendant cost and to improve long term software reliability. This item represents the first essential step in the standardization process (i.e., development of minimum standardization criteria). Standardization has many implications such as standard language design, standard operating system design, standard interface implementation and standard management practices; therefore, early development of criteria is mandatory. Standardized software is the easiest path to fault free software.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 4
3. REQUIRED LEVEL: 8 DATE 1979
4. ESTEMATE OF NON-NASA ADVANCEMENT: LEVEL 4

5. ALTERNATIVE SOLUTIONS:

None.

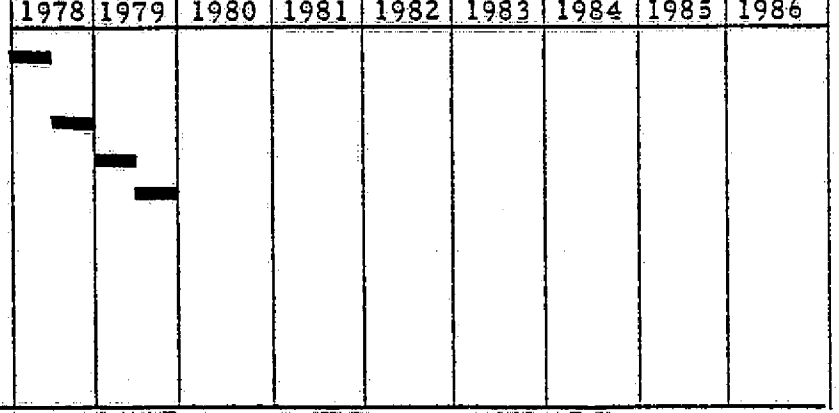
6. RELATED ACTIVITIES:

TI-06 AOL design guidelines/standards.
TI-08 Interface criteria for NASA distributed processor applications.
TI-17 Classification of error mechanisms.
TI-19 Software prototyping methods.

7. RECOMMENDED NASA EFFORT: Begin immediately an analysis of the full range of implications associated with standardization. Start development of criteria in mid 1978. Test the criteria by selecting candidate software packages and evaluating criteria validity in time to support accumulation of standard libraries in the 1980 - 1990 time frame.

ESTEMATED DEVELOPMENT COSTS: $150K - $500K PRIORITY: 4

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: AOL Design Guidelines/Standards NO. TI-06

1. DESCRIPTION OF ITEM: A shift to an increased emphasis on application oriented languages must be anticipated. This is due to the growing number of engineers (as opposed to professional programmers) developing or writing programs. Thus a requirement exists to develop the capability to quickly and economically define application oriented languages. As an initial step, NASA needs to produce AOL design guidelines and standards to ensure language consistency and compatibility with real-time distributed space systems.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 3
3. REQUIRED LEVEL: 6 DATE 1979
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

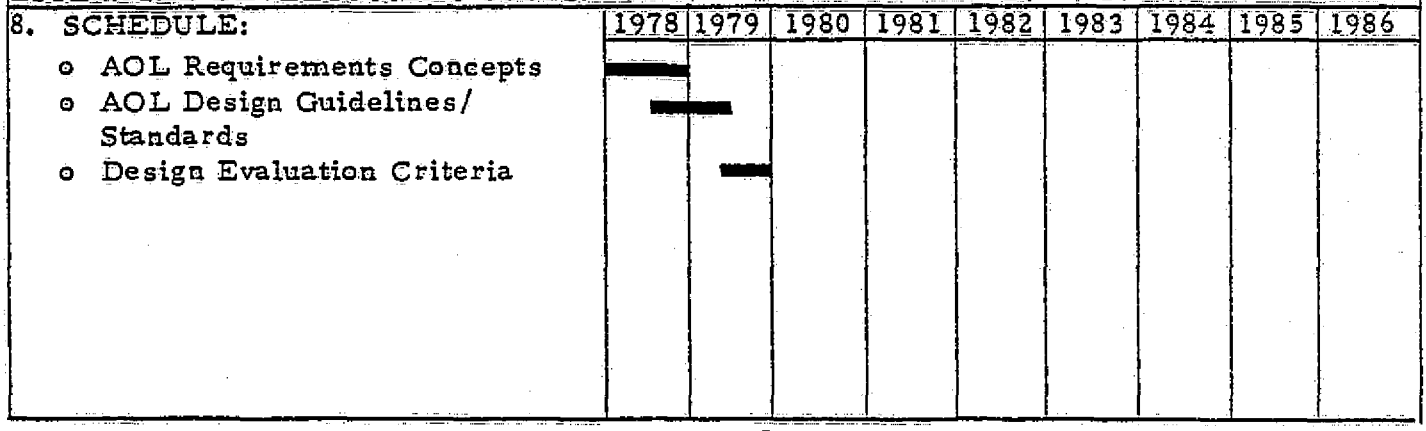
5. ALTERNATIVE SOLUTIONS:
   (1) Use assembly language for applications.
   (2) Use a variety of currently available HOL's.
   Both of these options are expensive; each leads to significant integration problems; and both result in less than desirable reliability.

6. RELATED ACTIVITIES:
   TI-01 Requirements decomposition and structuring guidelines.
   TI-05 Standardization criteria for NASA software.
   TI-07 AOL compiler generator cost factors.
   TI-08 Interface criteria for NASA distributed processor applications.

7. RECOMMENDED NASA EFFORT: Begin, in 1978, to determine concepts for the development of requirements of AOL's. Follow this with the development of AOL design guidelines/standards and design evaluation criteria. This effort should be scheduled such that the guidelines/standards will be available before 1980. This will allow time to produce an operational AOL generator prior to the period of heavy development from 1983 - 1990.

ESTIMATED DEVELOPMENT COSTS: $100K - $250K PRIORITY: 3

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: AOL Compiler Generator Cost Factors NO. TI-07

1. DESCRIPTION OF ITEM: A compiler is needed for each language which will be designed and for each computer the language will be used for. Although considerable research and development effort has been expended, compilers remain as very expensive items. If these costs are ever to be reduced, an understanding of what the costs are and why they are so high must be reached. This Technology Item recommends a NASA study effort to determine these AOL cost factors.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 3
3. REQUIRED LEVEL: 4 DATE 1978
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 3

5. ALTERNATIVE SOLUTIONS: None

6. RELATED ACTIVITIES:
TI-06 AOL design guidelines/standards.

7. RECOMMENDED NASA EFFORT: Initiate a study effort no later than calendar year 1978 to: (1) Identify representative AOL compiler generators, (2) Trace the development history of these generators, and (3) Perform a cost assessment of all major cost factors.

ESTIMATED DEVELOPMENT COSTS: $100K - $150K PRIORITY: 3

8. SCHEDULE:

	1978	1979	1980	1981	1982	1983	1984	1985	1986
o Typical AOL generators identification	▬								
o Development history tracing	▬								
o Cost assessment	▬								

COST/TIME/PRIORITY ASSESSMENT

TITLE: Interface Criteria for NASA Distributed Processor Applications NO. TI-08

1. DESCRIPTION OF ITEM: Virtually all future payloads will use distributed systems in one form or another. In addition, this concept will be the enabling function of intelligent instruments. System partitioning and interconnection techniques are particularly crucial in development of real-time systems. Any network of computers has, for the software as well as the hardware, the inherent problem of selecting interprocess communication techniques and selection of functions to be or not be distributed. This Technology item (developing interface criteria for NASA distributed processor applications) represents a critical first step toward solving these problems.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2
3. REQUIRED LEVEL: 6 DATE 1980
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 2

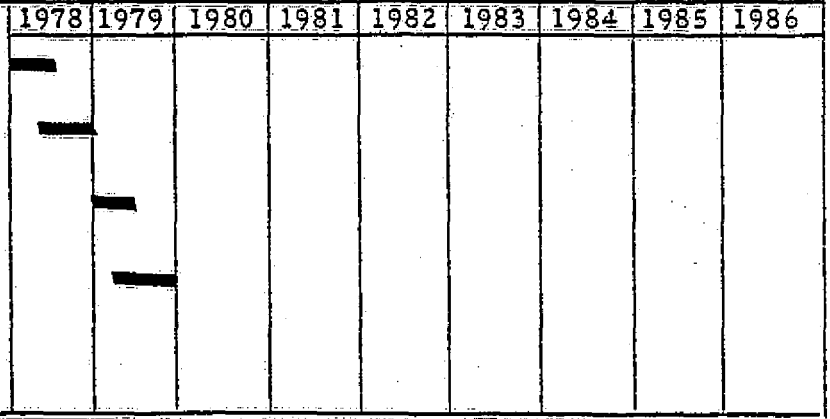
5. ALTERNATIVE SOLUTIONS: Use of high cost non-standard interfaces. This alternative poses unmanageable integration problems.

6. RELATED ACTIVITIES:
   - TI-01 Requirements decomposition and structuring guidelines.
   - TI-05 Standardization criteria for NASA software.
   - TI-09 Task control structures for distributed processor environments.
   - TI-10 Program organization methods for real-time fault recovery.
   - TI-12 Control structures for adaptive systems.

7. RECOMMENDED NASA EFFORT: Begin studies to identify potential configurations and characterize data and control flow. Perform trades to select interprocess communication techniques and establish interface criteria for NASA distributed systems. This effort should begin as soon as possible and be scheduled for completion before 1980.

ESTIMATED DEVELOPMENT COSTS: $100K - $200K PRIORITY: 10

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Task Control Structures for Distributed Processor Environments NO. TI-09

1. DESCRIPTION OF ITEM: Distributed systems (i.e., systems which contain multiple processors, each with its own executive, performing dedicated functions as part of a single partitioned system) are not new nor particulary novel. However, managing concurrent processes in real-time, space oriented distributed systems is new. The criticality of proper operation is much higher than for ground based systems. For this reason, NASA should begin to develop task control structures for distributed processors to simplify allocation, minimize communication and minimize interference.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 4
3. REQUIRED LEVEL: 8 DATE 1981
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

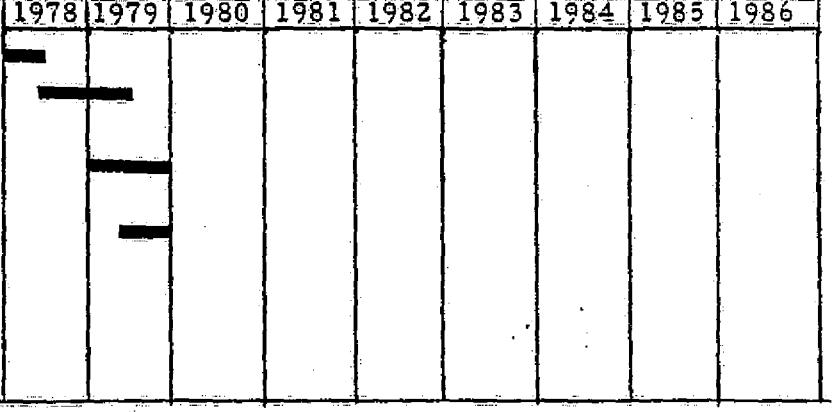
5. ALTERNATIVE SOLUTIONS: Adapt task control structures of ground based systems. Risk entails the potential loss of efficiency and reliability required for space operations.

6. RELATED ACTIVITIES:
   - TI-01 Requirements decomposition and structuring guidelines.
   - TI-05 Standardization criteria for NASA software.
   - TI-08 Interface criteria for NASA distributed processor applications.
   - TI-10 Program organization methods for real-time fault recovery.
   - TI-12 Control structures for adaptive systems.

7. RECOMMENDED NASA EFFORT: NASA should begin no later than calendar 1978 to identify techniques for real-time concurrent processes and follow this with research into task allocation and scheduling methods. These two items will provide the basis for developing task control structures for real-time distributed systems. The goal should be to have these task control structures in an operational state by 1980.

ESTIMATED DEVELOPMENT COSTS: $100K - $250K PRIORITY: 5

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Program Organization Methods for Real-Time Fault Recovery NO. TI-10

1. DESCRIPTION OF ITEM: Lack of effective recovery in real-time space activity can endanger space missions. In the past this has been compensated for by using sophisticated (and expensive) redundant hardware systems. The main problem is that it is difficult (if not impossible) to determine what happens between the time an error occurs and the time it is detected. The objective of this Technology Item is to develop methods of program organization that would maintain status in a way to allow backup and recovery with minimum loss of data and control in real-time systems.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 3
3. REQUIRED LEVEL: 6 DATE 1982
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

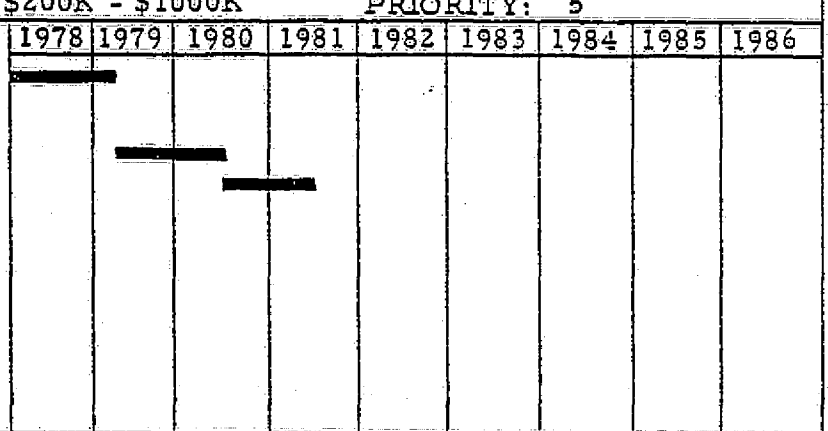
5. ALTERNATIVE SOLUTIONS:
   (1) Redundant systems with voting logic. (3) Acceptance of risk.
   (2) Fault free hardware/software.
   There are obvious drawbacks to each of these alternatives. Redundant systems are expensive and add significantly to power, weight and volume requirements. Fault free hardware/software has not been totally obtained and probably never will be. Risk may be acceptable for unmanned missions, but will presumably never be for manned spaceflight.

6. RELATED ACTIVITIES:
   TI-01 Requirements decomposition and structuring guidelines.
   TI-09 Task control structures for distributed processor environments.
   TI-12 Control structures for adaptive systems.
   TI-17 Classification of error mechanisms.

7. RECOMMENDED NASA EFFORT: NASA should begin research into the structure and organization of software systems designed to provide reliable fault recovery in real-time operations. The end goal is the development of appropriate program organization methods by 1982.

ESTIMATED DEVELOPMENT COSTS: $200K - $1000K PRIORITY: 5

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Adaptive High Speed Buffering Techniques for Dynamic Networks NO. TI-11

1. DESCRIPTION OF ITEM: Advanced buffering techniques are to be developed for catering to the high data rates expected in the dynamic environments of the 1980's. These techniques should be capable of adapting themselves to the characteristics of the data environments to achieve optimal utilization of the available resources.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2
3. REQUIRED LEVEL: 8 DATE 1983
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

5. ALTERNATIVE SOLUTIONS:
   - o Continued use of present day techniques is an alternative which may however gate the expected high volume data acquisition and processing required in the future missions.
   - o Significant orders of magnitude change in mass memory management might be another alternative.

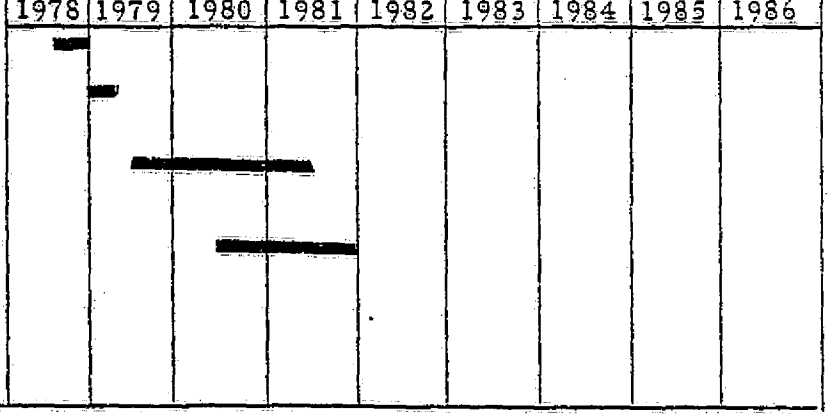
6. RELATED ACTIVITIES:

   None.

7. RECOMMENDED NASA EFFORT: The effort to be undertaken by NASA to meet the requirements is a study (to be initiated in 1978) to assess the state-of-the-art in buffering techniques and related developments in computer architecture and other hardware technologies followed by the development of adaptive techniques capable of optimal utilization of these resources in dynamic environments.

ESTIMATED DEVELOPMENT COSTS: $250K - $750K PRIORITY: 2

8. SCHEDULE:
   - o State-of-the-art assessment
   - o Impact of advancements in related areas
   - o New adaptive techniques development
   - o Implementation, testing and documentation



COST/TIME/PRIORITY ASSESSMENT

TITLE: Control Structures for Adaptive Systems NO. TI-12

1. DESCRIPTION OF ITEM: Control structures are needed to adaptively deploy available software to meet the requirements of the automated intelligence environments. This dynamic restructuring of the software to adapt it to the environment will enable NASA to plan and execute missions involving considerable autonomous capabilities.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2
3. REQUIRED LEVEL: 6 DATE 1981
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 3-4

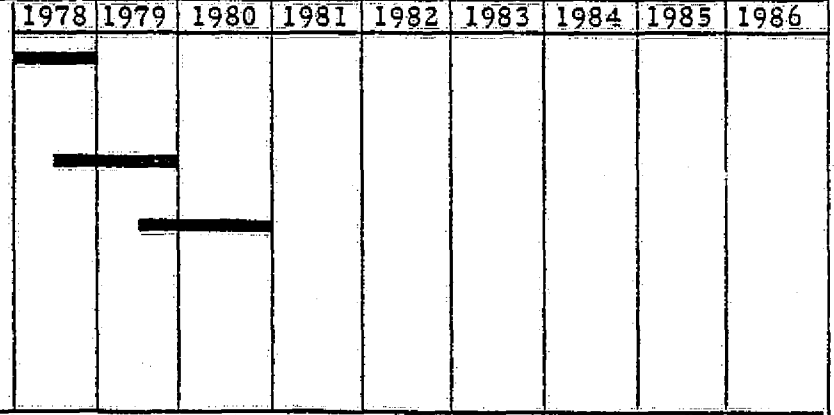
5. ALTERNATIVE SOLUTIONS: The alternative is a software system that has little adaptability to the changing environment and incapable of supporting autonomous unmanned missions involving at least rudimentary automated intelligence.

6. RELATED ACTIVITIES:
   - TI-04 Query guided implementation methods.
   - TI-05 Standardization criteria for NASA software.
   - TI-09 Task control structures for distributed processor environments.
   - TI-14 Adaptive search and sort procedures.
   - TI-15 Restructured image analysis software for parallel processing.

7. RECOMMENDED NASA EFFORT: NASA should initiate a study to develop appropriate techniques for real-time adaptive restructuring of software to meet the environmental requirements to ensure that the software technology stays ahead of the requirements of autonomously operated missions.

ESTIMATED DEVELOPMENT COSTS: $300K - $500K PRIORITY: 1

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Impacts of Natural Communication Methods on NASA Payload Systems

NO. TI-13

1. DESCRIPTION OF ITEM: It is essential for the efficient usage of available human resources onboard that natural communication methods be used to the extent feasible. Accordingly it is necessary to evaluate their influence and potential impact on the supporting payload software system and thereby ensure the success of the deployment of such methods. Towards this end, typical implementation of natural communication methods onboard for specific experimental tasks should be studied and the results used to derive an overall assessment of the impact of these methods on NASA payload software design.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2-3
3. REQUIRED LEVEL: 6 DATE 1981
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

5. ALTERNATIVE SOLUTIONS: Traditional methods of communication with the computers, when used exclusively, are not necessarily efficient in multiple task environments of future manned missions and are therefore a poor alternative to having natural communication methods as an added capability.

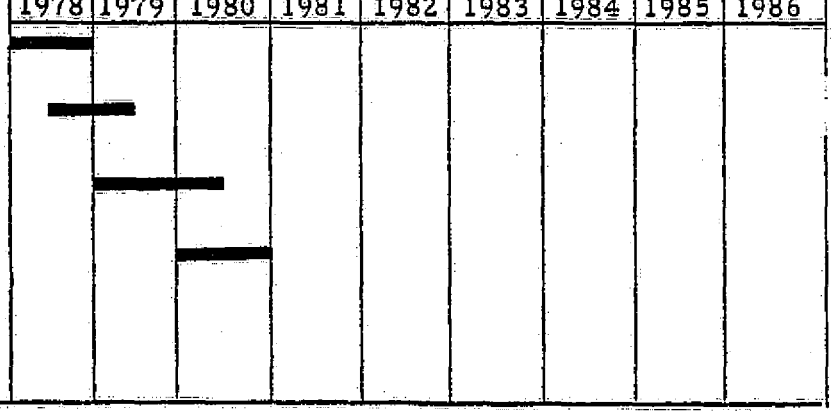
6. RELATED ACTIVITIES:

TI-08 Interface criteria for NASA distributed processor applications.

7. RECOMMENDED NASA EFFORT: NASA efforts in this area should include an ongoing monitoring of the developments in natural communication methods and technology to take advantage of such capabilities as and when this becomes practical. This be should be accompanied by an assessment of their impact on NASA software requirements and design.

ESTIMATED DEVELOPMENT COSTS: $150K - $250K PRIORITY: 1

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Adaptive Search and Sort Procedures NO. TI-14

1. DESCRIPTION OF ITEM: Adaptive search and sort procedures are required to ensure that the deployment of pattern recognition, image processing and related algorithms will be efficient and enable processing the high volume data in near real time as required in future NASA missions. These routines should be adaptive to the data characteristics of the arrays being manipulated.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2
3. REQUIRED LEVEL: 6 DATE 1981
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 3-4

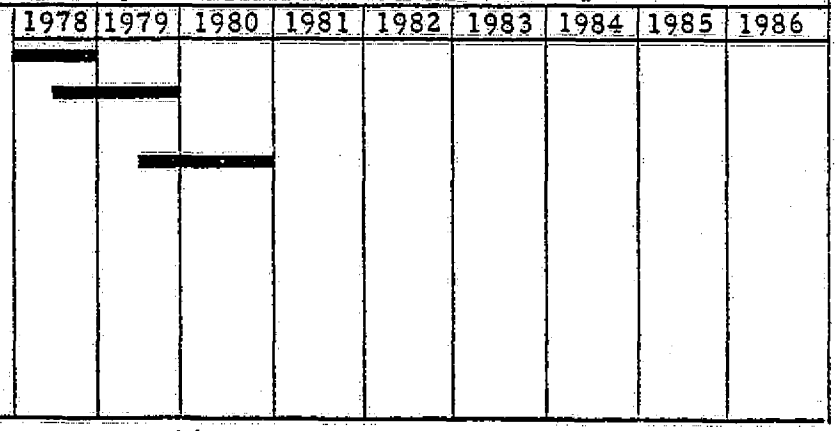
5. ALTERNATIVE SOLUTIONS: Use of non-standardized, arbitrarily picked search and sort routines currently available as parts of different pattern recognition and image processing packages, is an alternative involving duplication of software and inefficient routines not particularly suited to NASA requirements.

6. RELATED ACTIVITIES:
   - TI-05 Standardization criteria for NASA software.
   - TI-12 Control structures for adaptive systems.
   - TI-15 Restructured image analysis software for parallel processing.
   - TI-16 Optimal large array partitioning procedures.

7. RECOMMENDED NASA EFFORT: NASA should initiate a study to perform the requirement analysis for search and sort techniques in the context of image processing and pattern recognition activities and follow it up with development of appropriate adaptive search and sort routines to meet these requirements.

ESTIMATED DEVELOPMENT COSTS: $150K - $250K PRIORITY: 1

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Restructured Image Analysis Software for Parallel Processing NO. TI-15

1. DESCRIPTION OF ITEM: Onboard image processing/pattern recognition can be enabled by restructuring the key software functions to achieve maximum parallel processability. This can be attained by identifying the parallelism in these functions and adapting a suitable computer architecture which will take full advantage of the advancements in low cost hardware technology and parallel processing concepts.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2-3
3. REQUIRED LEVEL: 6 DATE 1981
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

5. ALTERNATIVE SOLUTIONS: Alternatives to restructuring image analysis software (for parallel processing) are:
   - o To use the software as is in non-parallel processing environment.
   - o To create new software expressly for parallel processing.
   - o Develop image processing dedicated hardware.

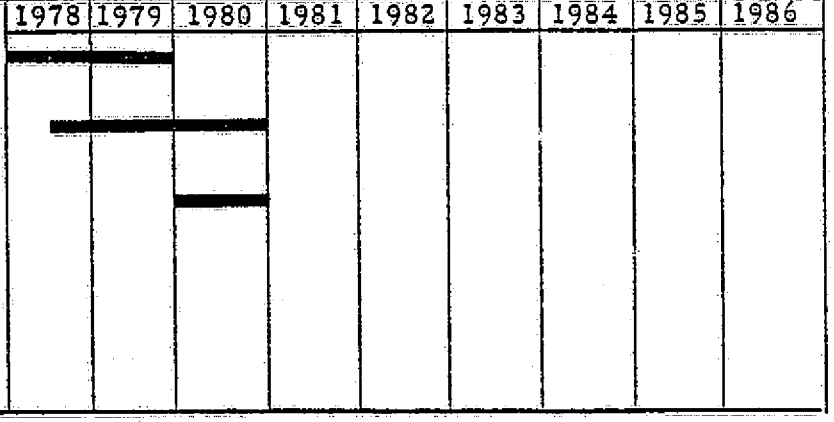
   The first two alternatives are either inefficient or unnecessarily expensive. The last needs further exploration.

6. RELATED ACTIVITIES:
   - TI-05 Standardization criteria for NASA software.
   - TI-09 Task control structures for distributed processor environments.
   - TI-12 Control structure for adaptive systems.

7. RECOMMENDED NASA EFFORT: NASA should direct its efforts towards enabling onboard image processing/pattern recognition functions through achievement of maximum parallelism in these processing activities. This is to be done through a study which will identify all the parallelism inherent in these functions not only in terms of parallel or concurrent identical processing on different pixels (data subsets) but also in terms of parallel execution of different computations on the same data.

ESTIMATED DEVELOPMENT COSTS: $250K - $1000K PRIORITY: 7

8. SCHEDULE:
   - o Parallel processible functions identification
   - o Restructuring/modification of image analysis
   - o Identification of computer architecture for implementing the restructured software system



COST/TIME/PRIORITY ASSESSMENT

TITLE: Optimal Large Array Partitioning Procedures NO. TI-16

1. DESCRIPTION OF ITEM: Optimal large array partitioning procedures offer a viable solution to the problems of large array manipulations such as matrix transpositions under the constraints of the processing system configurations expected onboard in the near future. Development of such procedures will enhance the capability of the processor in meeting the image processing requirements more effectively.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2-3
3. REQUIRED LEVEL: 6 DATE 1983
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 3-4

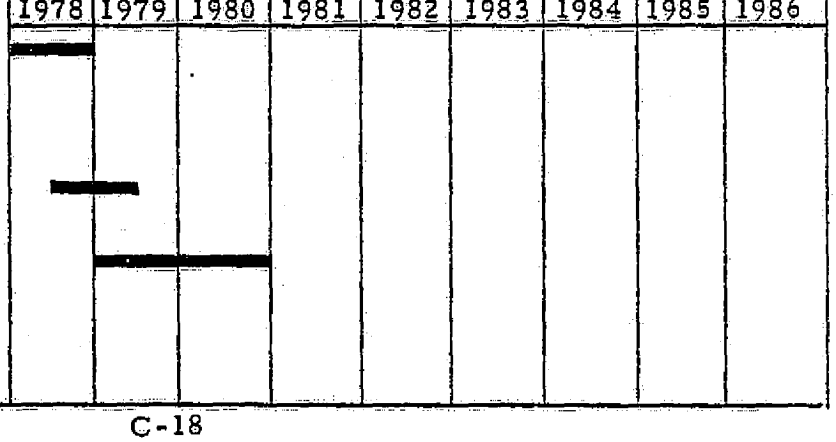
5. ALTERNATIVE SOLUTIONS: Electronic cyclic memories can alleviate to a considerable extent the problem of large array manipulations such as matrix transposition. This alternative has only been conceptually visualized. But, practical solutions for onboard implementation have yet to be realized.

6. RELATED ACTIVITIES:
   - TI-05 Standardization criteria for NASA software.
   - TI-12 Control structure for adaptive systems.
   - TI-14 Adaptive search and sort procedures.

7. RECOMMENDED NASA EFFORT: It is necessary that NASA should initiate a study of the feasibility and development of optional procedures for large 2 and 3 dimensional arrays manipulations to meet the needs of data compression, image processing scene analysis problems.

ESTIMATED DEVELOPMENT COSTS: $150K - $250K PRIORITY: 1

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Classification of Error Mechanisms NO. TI-17

1. DESCRIPTION OF ITEM: Study of past experience in software errors, their types, characteristics and frequency or likelihood helps in identifying the causes of such errors and leads to a reduction of the chances of their occurence in future missions. Accordingly classification of the error mechanisms will enhance the cause of fault free software development.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 2-3
3. REQUIRED LEVEL: 6 DATE 1980
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

5. ALTERNATIVE SOLUTIONS: Fault tolerant software development, if feasible, is an alternative. However, in general, the study of error mechanism will aid this alternative also in addition to reducing the likelihood of such errors.

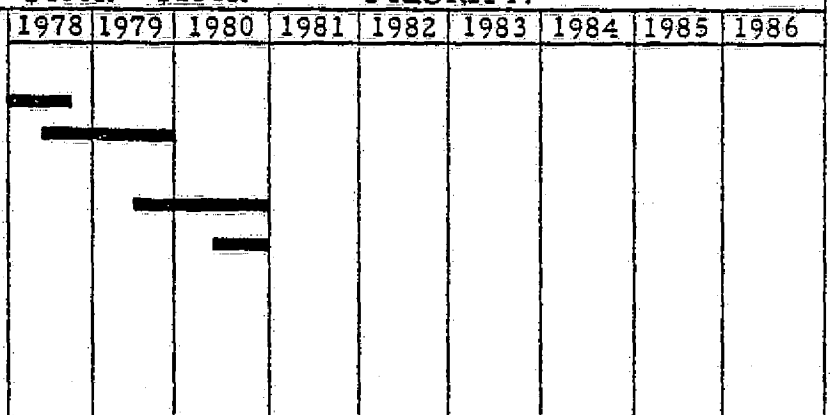
6. RELATED ACTIVITIES:

TI-10 Program organization methods for real-time fault recovery.
TI-19 Software prototyping methods.

7. RECOMMENDED NASA EFFORT: NASA efforts in this area should cover all recorded past software development activities to ensure that past experience will serve to enhance future performance. The tasks under this effort are as itemized below.

ESTIMATED DEVELOPMENT COSTS: $150K - $250K PRIORITY: 4

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Guidelines for Design Documentation Consistency NO. TI-18

1. DESCRIPTION OF ITEM: A major problem in the decomposition and restructuring of software design is the lack of a documentation methodology to accurately portray it. Many errors are introduced into programs during development, change and maintenance phases, because the programmer does not clearly understand the intent of the designer. This leads to much higher than required life-cycle costs and often results in less than reliable software. New methods of documentation, compatible with state-of-the-art design techniques, must be found.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 3-4
3. REQUIRED LEVEL: 8 DATE 1980
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 5

5. ALTERNATIVE SOLUTIONS:

None.

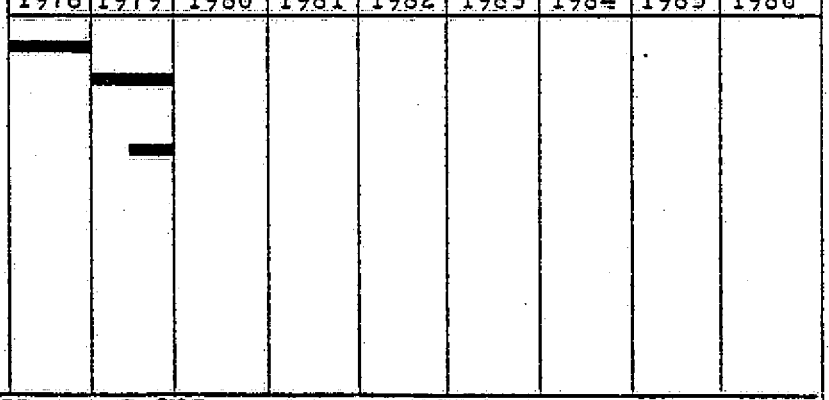
6. RELATED ACTIVITIES:

TI-01 Requirements decomposition and structuring guidelines.
TI-05 Standardization criteria for NASA software.
TI-06 AOL design guidelines/standards.

7. RECOMMENDED NASA EFFORT: NASA should begin no later than calendar 1978 to develop methods of obtaining documentation consistency and from these methods generate guidelines to ensure consistency of future software development projects. In this way, documentation which clearly and concisely describes the problem and its solution, and which reflects program structure will be enabled.

ESTIMATED DEVELOPMENT COSTS: $100K - $250K PRIORITY: 3

8. SCHEDULE:



COST/TIME/PRIORITY ASSESSMENT

TITLE: Software Prototyping Methods NO. TI-19

1. DESCRIPTION OF ITEM: Concern for the high cost of design errors during and after software development has led to a search for ways to proof a design before committing it to code. One of the more promising techniques recently receiving increased emphasis is software prototyping. Software prototyping is a method of testing the key elements of a design at minimal cost, prior to starting development of the operational system. This technique has been successfully used on a few ground based systems, but has yet to be applied to real-time space systems. The key to this Technology Item is development of true, low-cost prototyping techniques.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL 3
3. REQUIRED LEVEL: 6 DATE 1980
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL 4

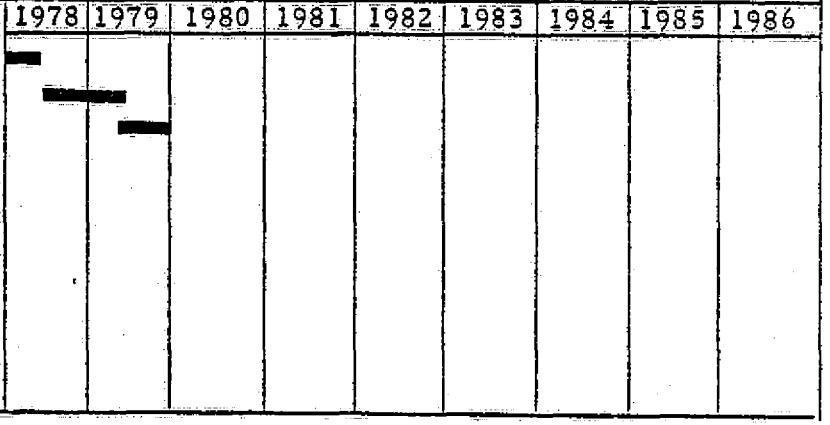
5. ALTERNATIVE SOLUTIONS:
   (1) Requirements-to-code translation.
   (2) Consistent decomposition and structuring methods.
   (3) The two alternatives, although highly desirable goals, do not fully satisfy the problem, since neither gives a good indication of the feasibility or cost of committing the design to code, regardless of the quality of the design.

6. RELATED ACTIVITIES:
   TI-03 Simplified software development system demonstration model.
   TI-05 Standardization criteria for NASA software.

7. RECOMMENDED NASA EFFORT: NASA should begin research and development of real-time software prototyping. The cost savings on large development projects through ensuring design feasibility will far outweigh the cost of developing proofing methods and will lead to more reliable end products.

ESTIMATED DEVELOPMENT COSTS: $100K - $250K PRIORITY: 4

8. SCHEDULE:
   - o Technology analysis
   - o Low-cost prototyping methods
   - o Evaluation criteria



COST/TIME/PRIORITY ASSESSMENT

TITLE: Software Technology Monitoring NO. TI-20

1. DESCRIPTION OF ITEM: Software technology is continually advancing, and though growth is primarily evolutionary, there is nothing to preclude a revolutionary breakthrough, particularly in view of the dramatic advances in hardware technology. NASA should be in a position to capitalize on all technology advances, whether achieved by government, in Universities, or in industry. It can best do this by establishing a mechanism to continually review state-of-the-art technology.

2. CURRENT STATUS OF TECHNOLOGY: LEVEL N/A
3. REQUIRED LEVEL: N/A DATE 1978-1990
4. ESTIMATE OF NON-NASA ADVANCEMENT: LEVEL N/A

5. ALTERNATIVE SOLUTIONS:

None.

6. RELATED ACTIVITIES:

None.

7. RECOMMENDED NASA EFFORT: NASA should establish a function to continually monitor state-of-the-art software technology. This function should project NASA needs, establish milestones for industrial advancement, and identify those areas where NASA must force progress. In this way, NASA will get full benefit of advances outside of the organization and at the same time be in a position to ensure that its requirements are met, at the time needed.

ESTIMATED DEVELOPMENT COSTS: $50K per year (1977 dollars) PRIORITY: 5

8. SCHEDULE: