FINAL ENGINEERING REPORT

ETE RESTRUCTURING PROJECT, PHASE I

PREPARED BY

M. ANDREW LIPSCOMB
ENGINEERING EXPERIMENT STATION
GEORGIA INSTITUTE OF TECHNOLOGY

PREPARED FOR

WARNER ROBINS AIR LOGISTICS CENTER
ROBINS AIR FORCE BASE
F09603-78-G-4368-0004
FOREWORD

The ETE Restructuring project under Contract F09603-78-G-4368, Tasks 0004 and 0015, was conducted by the Engineering Experiment Station at Georgia Tech. This report is the Final Report for Task 0004, administered under Georgia Tech Project A-2259 by the EW System Software Branch of the Defense Systems Division, Systems Engineering Laboratory.

The Georgia Tech Project Director was Mr. M. Andrew Lipscomb. The project was under the direct supervision of Mr. Thomas M. Miller, division chief, and the general supervision of Mr. R. P. Zimmer, laboratory director. Mr. Harry Jennings was the project monitor at Warner Robins Air Logistics Center.
# TABLE OF CONTENTS

1.0 Summary &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs...
1.0 Summary

This report describes a study conducted by the Georgia Tech Engineering Experiment Station for the United States Air Force, Warner Robins Air Logistics Center (WR-ALC) to determine the technical and economic feasibility of rewriting an existing computer program to produce a functionally equivalent program in a more maintainable form. Maintainability, as applied to computer software, refers to the ease with which a program can be modified in response to changing requirements of the program user. WR-ALC has responsibility for such maintenance on a variety of Electronic Warfare (EW) systems. Software engineering techniques, such as structured programming and modular design, are available for enhancing software maintainability; it is often assumed, however, that such techniques tend to degrade software efficiency in terms of core memory utilization and execution. In order to test this assumption, WR-ALC engineers outlined the following task to be performed by Georgia Tech:

1. Develop procedures for restructuring an existing computer program to produce a functionally equivalent program of higher maintainability.

2. Apply these procedures to the Emitter Trackfile Entry (ETE) module of the AN/ALR-46 Operational Flight Program as a test case.

3. Determine differences in core memory utilization and execution time between the original and restructured versions.

In addition to these technical points, EES was required to report the effort expended in the restructuring project in terms of a work measure which may be employed to predict level of effort vs. program size for restructuring tasks of similar scope. This measure is for use by WR-ALC in determining the economic feasibility of restructuring in an operational setting.

The following results were obtained:

1. Core memory utilization in the restructured version increased by approximately 25%.
2. Execution time for the restructured version decreased by an average of approximately 30%.

3. The effort required, stated in terms of lines of original code per man-day was:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>For the specification and analysis</td>
<td>5.8 lines/day</td>
</tr>
<tr>
<td>For the development and validation</td>
<td>9.6 lines/day</td>
</tr>
<tr>
<td>Total Effort</td>
<td>3.6 lines/day</td>
</tr>
</tbody>
</table>

A large portion of the increase in core memory utilization is attributable to the policy adopted for purposes of the study, of providing maximum separation of function in the software design in accord with the goal of high maintainability. In an operational environment, a more balanced design could yield an acceptable level of functional separation without such a large penalty in core memory. In view of the substantial decrease in execution time, it is concluded that restructuring is a technically feasible option for increasing maintainability.

The level of effort required compares favorably to the effort required to develop new software from scratch. The development and validation cost is approximately the same as the accepted industry average. Industry figures on the cost of requirements analysis and specification for new software are difficult to obtain and interpret, but it is unlikely that a specification of comparable scope and detail could be developed with less effort than was required for the restructuring. Furthermore, substantial portions of the analysis required for restructuring can be conducted by persons with little prior experience in EW software. Restructuring by inexperienced personnel can decrease the monetary cost of the effort and does not divert experienced personnel from the ongoing maintenance efforts on the operational software.

On the basis of these and other conclusions it is recommended that restructured software be given a field trial to determine more precisely the extent to which maintainability is enhanced. Studies aimed at determining the technical feasibility of conversion to a higher order language, improving system reliability, and formulating long range maintenance plans are also recommended.

2.0 Project Documentation

The ETE restructuring project encompassed a number of subtasks which
have been documented in separate reports. These are listed below with a summary of the contents of each. The present report contains the results of the efficiency comparisons, the level of effort report, and various conclusions and recommendations drawn from the test results, and the experience of the project team in conducting the work. The reports listed below were prepared under two tasks on Contract No. F09603-78-G-4368. The task number is given in parentheses following the report title.

- **CPCI Computer Program Development Specification for the Emitter Trackfile Entry Module of the ALR-46 Operational Flight Program (0004).**
  
  This document contains the functional specification for the restructured ETE module. It contains a description of the functional requirements on ETE, and defines the principle modules in the implementation.

- **CPCI Computer Program Product Specification for the Emitter Trackfile Entry Module of the ALR-46 Operational Flight Program (0004).**
  
  This document contains the detailed design specification for the restructured ETE module. The specification is organized on the basis of the program modules and includes, for each module, a functional summary, a list of data accessed and modified, and a Warnier-Orr chart illustrating the module's organization.

- **CPCI Computer Program Development Test Plans/Procedures (0004).**
  
  This document describes the procedures and test environment used in the execution time comparison tests. It describes the driver software developed to exercise ETE and defines the benchmark test cases to be used in the testing.

- **Final Engineering Report, ETE Restructuring Project, Phase II (0015).**
  
  This report documents the procedures developed for the restructuring and benchmark test cases used in the comparative testing. The test cases are related to specific functions in ETE. Execution time results are given in each case.

### 3.0 Effect of Restructuring on Program Efficiency

The results of the core memory and execution time efficiency compari-
sons are provided in Table 1.

The execution time figures in Table 1 give the range of differences measured between the two versions and the mean difference for the 23 benchmark tests (equally weighted). As described in the test case documentation, Task 0015 Final Engineering Report, timing accuracy in the execution time measurements were limited by instabilities in the system real time clock. The high value reported for each figure represents a best estimate. Test results for individual cases are reported in the test case documentation.

4.0 Level of Effort

Table 2 shows the time in hours and, where applicable, hours per line of original code devoted to the ETE restructuring task, broken down by subtask and by category of personnel contributing to the effort. The figures represent only the time devoted specifically to technical aspects of the project. Administrative tasks are excluded. Time spent in the preparation of documents is included for documents, such as the specifications, which are part of the permanent documentation of the software. Time spent in the preparation of other documents, such as the Final Engineering Report, which are not normally part of a software development effort is excluded. The figures include technical services, such as drafting, but exclude secretarial services.

The total effort is broken into four subtasks: Software Development, Development of Procedures, Development of Test Driver, and Evaluation.

Software Development is further decomposed into Functional Specification, Design Specification, and Coding and Validation. The Functional Specification and Design Specification subtasks include all the analysis efforts expended in the preparation of the documents. In particular, the Functional Specification subtask includes time spent in the review of existing documentation and the initial analysis of the software being restructured. The Coding and Validation subtask includes time spent in the preparation of documentation relevant to this phase of the effort.

Development of Procedures includes all time spent in researching and documenting the Restructuring Procedures. The time required in this effort is independent of the scope of the restructuring, thus the "hours per line" figure is not applicable.

Development of the Test Driver includes only the fixed effort asso-
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum improvement</td>
<td>13%</td>
</tr>
<tr>
<td>Maximum improvement</td>
<td>49%</td>
</tr>
<tr>
<td>Mean improvement</td>
<td>30%</td>
</tr>
</tbody>
</table>

Table 1. Improvement in Execution Time in Restructured ETE
associated with this task, for example, the initial design and development effort and the effort expended in modifying the driver to perform timing measurements. Effort expended in performing the minor modifications required for each run of the driver is included, as applicable, with the Coding and Validation or Evaluation figures.

The Evaluation subtask includes the time spent developing the Comparative Test Plan and test data, and the time spent in conducting the comparative tests.

The categories of personnel listed are Engineers, Graduate Research Assistants and undergraduate Student Assistants. Graduate Research Assistants normally performed at the level of professional programmers, and Student Assistants performed technical support tasks such as drafting and data gathering.

5.0 Conclusions

The following subsections contain the principle conclusions drawn from the restructuring study. These conclusions are drawn from both the specific data presented in the previous section, and the general experience of the project team in performing the restructuring.

5.1 Technical Feasibility

The observed increase in core memory utilization is, in part, attributable to the relative priority of design goals adopted by the project team for purposes of the study. The project Statement of Work requires the production of a restructured program of equivalent specification. Since the objective of the project was primarily to evaluate the effects of structured, modular programming, the project team adopted a somewhat conservative interpretation of this requirement. Design changes in the restructured program were purely structural and organizational in nature; there were no changes in the basic processing algorithms implemented in the program.

At the detailed design level, design goals were stated in terms of functional separation (a maintainability characteristic), execution time efficiency, and core memory efficiency. Design tradeoffs arise most frequently from conflicts between requirements for functional separation and core memory efficiency. In keeping with the project objective of enhanced maintainability, functional separation was given highest priority. Since
<table>
<thead>
<tr>
<th></th>
<th>Engineer</th>
<th>Graduate Research Assistant</th>
<th>Student Assistant</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hours</td>
<td>hrs./line</td>
<td>hours</td>
<td>hrs./line</td>
</tr>
<tr>
<td>I. Software Development Total</td>
<td>1470</td>
<td>1.12</td>
<td>930</td>
<td>.71</td>
</tr>
<tr>
<td>A. Functional Specification</td>
<td>720</td>
<td>.55</td>
<td>690</td>
<td>.53</td>
</tr>
<tr>
<td>B. Design Specification</td>
<td>350</td>
<td>.27</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C. Coding and Validation</td>
<td>400</td>
<td>.30</td>
<td>240</td>
<td>.18</td>
</tr>
<tr>
<td>II. Development of Procedures</td>
<td>240</td>
<td>N/A</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>III. Development of Test Driver</td>
<td>40</td>
<td>N/A</td>
<td>70</td>
<td>N/A</td>
</tr>
<tr>
<td>IV. Evaluation</td>
<td>180</td>
<td>.09</td>
<td>40</td>
<td>.03</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1930</td>
<td>N/A</td>
<td>1040</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 2. Summary of Effort Expended
core memory efficiency was least compatible with this primary goal, it was

given lowest priority. Execution time efficiency was given secondary prio-

rity.

It is reasonable to expect that, in a more balanced approach, the
increase in core use could be lessened somewhat, although probably not
eliminated, without incurring substantial penalties in either execution time
or maintainability.

In view of the decrease in execution time in the restructured version
it is safe to conclude that restructuring is a technically feasible option for
increasing software maintainability.

5.2 Economic Feasibility

Conclusions regarding the economic feasibility and desirability of
restructuring must ultimately be drawn by WR-ALC personnel, but several
points from the study are noteworthy in this regard.

The economics of restructuring actually involves two questions.
First, is any sort of broad scope revision desirable? Secondly, if a major
revision is to be undertaken, what is the appropriate approach? The current
study does not address the question of economic desirability of a revision, as
such. This question will require determination of the relative cost of
maintaining revised software as opposed to continuing in the current operat-
ing mode. Tentative conclusions can be drawn, however, with regard to the
relative economy of restructuring versus other approaches to revision.

The most frequently suggested alternative to restructuring is re-
developing; that is, developing new software directly from the requirements
without reference to the existing code. Any software development effort
requires two phases: a requirements analysis phase terminating in a function-
al specification for the software and a design and development phase wherein
the functional specification is translated into operational code. Re-
structuring differs from redeveloping only in the requirements analysis
phase; given an adequate functional specification the design and development
phase is the same for both.

Any attempt to develop a thorough functional specification without
reference to the existing code will necessarily require an extensive commit-
ment of time by personnel with an intimate knowledge of the system's develop-
ment history. It has been the project team's experience that existing published documentation (as opposed to the private notes of the system engineers) is inadequate for software development. Restructuring, on the other hand, requires very little commitment of time from experienced personnel. None of the Tech personnel involved in the restructuring project had prior experience on the ALR-46 software. Although the project team frequently required consultation with WR-ALC engineers, the interchange usually consisted of focused question and answer sessions. The commitment of time by WR-ALC engineers was small compared to the total effort.

Since the diversion of experienced personnel is expensive both directly and in terms of the effort lost in the normal activities of operational software maintenance, it is safe to conclude that any revision effort should begin with analysis of the existing code. This is not to say the requirements analysis should not extend beyond the existing code, but the restructuring team will be better prepared for such extended analysis and efficient interchange with other personnel after an initial analysis of the existing code.

5.3 Effects of Scope on the Restructuring Effort

A significant portion of the ETE code performs operations which are more closely related to the function of modules external to ETE than to that of ETE itself. This sort of functional mixing is likely to be found in any program with a long history of modification. This being the case, broad restructuring efforts are likely to yield greater benefits in efficiency and maintainability than narrow ones.

On the other hand, as the current study indicates, restructuring at the level of primary modules can yield a significant payoff. Since piecemeal restructuring requires fewer personnel and implies less disturbance to the ongoing task of maintaining operational software, this approach may be preferable, at least in the early stages of a revision effort. A significant result from the current study is that the bulk of the effort in the restructuring lies in the requirements analysis phase. If this analysis is performed well initially, and design documentation is maintained during operational maintenance, then further revisions at a later date will be less costly.

5.4 Adequacy of the Restructuring Procedures

In the development of procedures for the restructuring, the principle
goal was to define design, testing, and documentation procedures which would remain valid in an operational setting. The procedures which evolved satisfy most requirements of good practice, that is, they provide sound design information and traceability of requirements. Nonetheless, they should be considered as prototypical rather than final. Experience in actual operations will undoubtedly reveal opportunities for providing better information, or providing it more efficiently. In particular, some of the more burdensome activities associated with documentation development are subject to automation.

5.5 Reliability Effects of Restructuring

Restructuring can lead to significant gains in software reliability. The team's experience indicates that reliability is enhanced in three ways. First, the in-depth analysis required to produce the Functional Specification provides a means of detecting problems in the existing design. A number of design anomalies were discovered in the original version of ETE and resolved in the restructuring.

Secondly, the design procedures employed in the effort lead to a high level of programmer efficiency (which is defined here as inversely proportional to the rate at which a programmer or designer introduces errors into the software). The great majority of modules passed validation testing in no more than three runs. A significant portion of the modules passed in one run, that is, no errors were detected.

Thirdly, the structured design procedures allowed the formulation of explicit test plans and test coverage measures for the detection of those errors which were introduced. The formal testing procedures, coupled with an informal review process, were highly effective in eliminating errors. It is impossible to say with certainty what percentage of the total errors in the program were detected since the number of undetected errors is not known. It is noteworthy, however, that of the total errors detected, including those that were detected in the comparative testing which effectively constituted a separate (and thorough) integration test, approximately 80% were detected in the initial validation testing and review process conducted at the module level.
Although 80% error detection at the module level is a good record for an initial implementation, it falls short of the reliability level required in operational EW software, and improved test coverage is indicated. It should be stressed that, while the test coverage requires refinement, the test procedures, which require the demonstration of a given level of coverage based on the program design, proved adequate. Without a structured design on which to base test plans, a level of test coverage is difficult to specify, and its effectiveness cannot be evaluated.

5.6 Psychological Factors

A recurring problem in evaluating the effectiveness of programming procedures is the separation of effects due to the procedures as such from effects due to the software development team implementing the procedures. In the present case, all contributors to the effort were aware that the intent of the project was to compare the performance of structured and unstructured programs; since most of the persons involved had a substantial professional commitment to structured design techniques, there was considerable motivation to produce high performance code. Furthermore, this motivation was purely personal. Since the task was performed as an experiment and, at least initially, the structured code was expected to perform less well than the original, there was no formal requirement to produce a given level of performance. Nonetheless, the programming team adopted (of their own volition) a process of independent review of all code produced. This review process was possibly as effective as the formal procedures in increasing reliability and efficiency.

While psychological factors bolstered the procedures, the reverse was also true. The project team viewed the procedures, including the testing, as a means of increasing their own effectiveness. There is an unfortunate tendency to view formal methods (particularly testing) as a club with which to belabor the software development team. In fact, no formal procedures can force high performance; good procedures allow high performance. Testing, both validation and performance testing, can be particularly valuable to programmers in providing immediate and specific feedback. Such feedback considerably speeds up the learning process. It is significant that toward the end of the coding and validation phase of the current effort, most modules
were passing validation testing on the first attempt.

5.7 Potential Benefits of Higher Order Language

It is widely accepted that the use of a higher order language can increase software maintainability and reliability. Although the current study did not address this assumption directly, it is consistent with the project team's experience. For instance, approximately 75% of the errors detected during testing were directly attributable to the vagaries of assembly language programming.

A more significant result in this regard is that large scale revision of software can produce substantial gains in efficiency. Given this finding, it is conceivable that conversion to a higher order language, in conjunction with restructuring, would be technically feasible.

6.0 Recommendations

As noted previously, the current study demonstrates that restructuring is technically feasible, and economically competitive with other means of large-scale revision. The exact benefits deriving from such revision can only be determined in the field. Similarly, the procedures employed require refinement through field experience. Thus the principle recommendation is for a trial fielding of restructured software. This fielding should be accompanied by additional studies and the formulation of a long-range plan for the maintenance process. These recommendations are described in greater detail in the following subsections.

6.1 Development and Fielding Efforts

A restructured section of software, including but not necessarily limited to ETE, should be installed in the ALR-46 OFP. This test case should be used to collect information on the relative ease of maintenance for the restructured software, and the effectiveness and efficiency of the design, coding, and validation procedures. This effort should be accompanied by the development of automated tools to aid the design and documentation process.

6.2 Additional Studies

A study should be conducted to evaluate the technical feasibility of converting to a higher order language such as FORTRAN or PASCAL. Ideally,
this effort should be conducted concurrently with the development of the restructured software for the field test mentioned previously. The bulk of the restructuring effort is in the functional specification phase. Once this is completed, both an assembly language (for installation) and a higher order language program (for testing) could be developed from the same specification with comparative ease. The assembly language and higher order language programs should then be tested to determine the relative efficiency of the two versions. Various levels of hybridization (assembly and higher order language mix) should also be evaluated.

The efficiency study should be accompanied by a reliability study aimed at developing more effective test coverage strategies.

6.3 Long-Range Planning

The results of the efforts above should be used in the formulation of a long-range plan for OFP revision and maintenance. Within the near future, changes in core memory technology (through the installation of electrically alterable read only memory) can be expected to remove the last absolute barrier to the use of higher order language. In the same general time period, the DOD language, Ada will become available. The decision on whether or not to convert to should be based on sound knowledge regarding the potential benefit of a higher order language.

If such conversion is undertaken a global revision of the OFP will be indicated. Such a revision, to be maximally beneficial, should employ procedures which have been thoroughly tested in the field and shown to be effective in producing high maintainability and reliability.

Long-range planning efforts should not be limited to the ALR-46. Structured design, particularly if accompanied by conversion to a higher order language, offers the potential for cutting maintenance costs by taking advantage of cross-system commonalities. Long-range planning should include the definition of a system taxonomy which delineates common functions among the various operational systems. In a global revision, such information can be used to design systems requiring less redundancy in development and maintenance efforts.