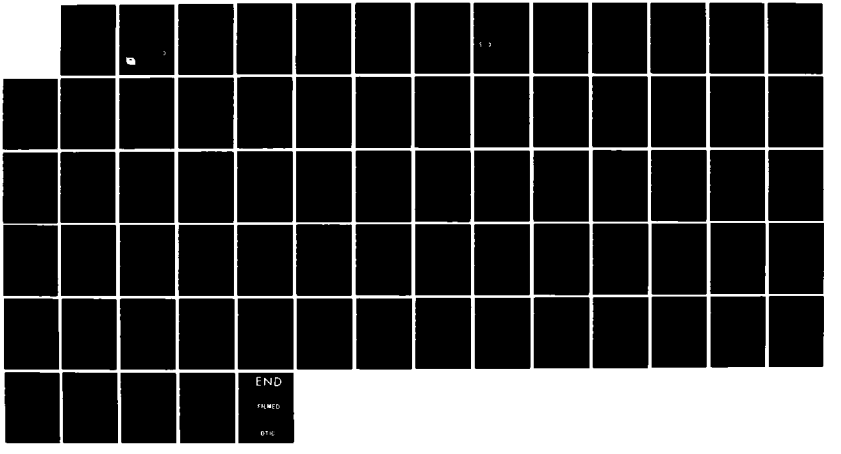
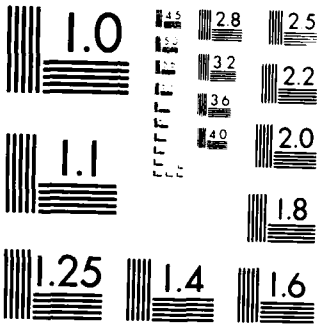
AD-A149 436 ADA (TRADEMARK) JOINT PROGRAM OFFICE OBJECTIVES AND PROGRESS -- THROUGH 1. (U) INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA J F KRAMER ET AL. SEP 84 1/1

UNCLASSIFIED IDA-M-22 IDA/HQ-84-29112 MDA903-84-C-0031 F/G 9/2 NL





AD-A149 436

(2)

IDA MEMORANDUM REPORT M-22

ADA* JOINT PROGRAM OFFICE OBJECTIVES AND PROGRESS — THROUGH 1983

John F. Kramer
Catherine W. McDonald

September 1984

Prepared for
Office of tne Under Secretary of Defense for Research and Engineering

INSTITUTE FOR DEFENSE ANALYSES

IDA Log No. HQ 84-29112

The work reported in this document was conducted under contract MDA 903 84 C 0031. The publication of this IDA Memorandum Report does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that agency.

This Memorandum Report is published in order to make available the material it contains for the use and convenience of interested parties. The material has not necessarily been completely evaluated and analyzed, nor subjected to IDA review.

Approved for public release; unlimited distribution.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD A149436	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Ada Joint Program Office Objectives and Progress - Through 1983		5. TYPE OF REPORT & PERIOD COVERED FINAL 6. PERFORMING ORG. REPORT NUMBER IDA Memorandum Report M-22
7. AUTHOR(s) John F. Kramer, Jr. Catherine W. McDonald		8. CONTRACT OR GRANT NUMBER(s) MDA 903 84 C 0031
9. PERFORMING ORGANIZATION NAME AND ADDRESS Institute for Defense Analyses 1801 N. Beauregard Street Alexandria, VA 22311		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task T-4-222
11. CONTROLLING OFFICE NAME AND ADDRESS Ada Joint Program Office, OUSDRE(R&AT) 400 Army-Navy Drive, 9th Floor Arlington, VA 22202		12. REPORT DATE September 1984 13. NUMBER OF PAGES 68
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DoD-IDA Management Office 1801 N. Beauregard Street Alexandria, VA 22311		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Ada Ada Joint Program Office, Life-Cycle, Methodology, Environments, APSE, DoD, software, human resources, reliability, adaptability, portability, STARS, METHODMAN, STEELMAN, STONEMAN

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In 1975, the Department of Defense (DoD) began the process of standardizing the high order languages used to write software for embedded computers. The first step was to form a High Order Language (HOL) Working Group to identify DoD's requirements for computer programming languages, evaluate the existing languages, and recommend the implementation and control of a "minimal set". Although an interim set of languages was established, none were considered to be a long term solution to the DoD's programming (cont.)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Continued

needs. Therefore, an international request for proposals was issued for a new common language. The final requirements document, STEELMAN, served as the standard for the selection of the new DoD HOL. In 1979, this new language, developed by Cii-Honeywell Bull, was named Ada in honor of Augusta Ada Byron, the countess of Lovelace.

On 12 December 1980, the Under Secretary of Defense for Research and Engineering established the Ada Joint Program Office to manage the DoD's effort to implement, introduce and provide life-cycle support for Ada. This paper outlines, past, present and future objectives of the Ada Joint Program Office.

IDA MEMORANDUM REPORT M-22

ADA* JOINT PROGRAM OFFICE OBJECTIVES AND PROGRESS — THROUGH 1983

John F. Kramer
Catherine W. McDonald

September 1984

INSTITUTE FOR DEFENSE ANALYSES
1801 North Beauregard Street
Alexandria, Virginia 22311

Contract MDA 903 84 C 0031
Task T-4-222

TABLE OF CONTENTS

		Page
PREFACE		v
1.0	LANGUAGE STANDARDIZATION	1
1.1	Standard Ada Language	1
1.1.1	Language Reference Manual	2
1.1.2	Military Standard	3
1.1.3	ANSI Standard	3
1.1.4	FIPS	3
1.1.5	ISO Standardization Activities	4
1.1.6	NATO	5
1.2	Ada Board	5
1.3	Compiler Conformance	8
1.3.1	Ada Validation Organization	8
1.3.2	Ada Compiler Validation Capability	9
1.4	Activities Related to Language Standard	10
1.4.1	Implementor's Guide	11
1.4.2	Operation Semantic Definition	12
1.4.3	Formal Semantic Definition	12
1.4.4	Replacement of Ada	13
2.0	TECHNOLOGY	15
2.1	Environment	17
2.1.1	Ada Programming Support Environment Requirements	17
2.1.2	Tool Portability	18
2.1.3	Conventions and Standards	21
2.1.3.1	Definitions	21
2.1.3.2	Tool Development	23
2.1.4	E&V Effort	26
2.2	Methodology	26
2.2.1	Requirements	27
2.2.2	Life-Cycle Methodologies	27
2.2.3	DoD Standards	28
2.3	Reusable Software	28
2.3.1	Standard Packages/Extensions	29
2.3.1.1	General Purpose Library	29
2.3.1.2	Ada Mathematics Library	30
2.3.2	Reusable Software Distribution	30
2.3.2.1	Develop Requirements	30
2.3.2.2	Distribution Facilities	31
2.3.2.3	Facility Operation	31
2.3.3	Tools	31
2.4	Run-Time Environments	32

3.0 HUMAN RESOURCES 33
3.1 Strategic Planning for Education and Training 34
3.2 Human Resource Management 35
3.3 Information Dissemination 36
3.4 Professional Community Development 37

4.0 ADA BUSINESS ENVIRONMENTS 39
4.1 Business Environment Strategies 41
4.1.1 Develop Organizational Strategies 41
4.1.2 Develop Hardware Configuration Strategy 43
4.1.3 Develop Application Area Strategy 44
4.2 DoD Software Policies 46
4.3 Facilitate Use of Ada 47
4.3.1 Develop Information Exchange Strategy 47
4.3.2 Information Dissemination 48
4.3.3 Cultivate Affinity Groups 49
4.3.4 Ada in Research 51
4.4 Incentives 52
4.4.1 Incentive Strategies 52
4.4.2 Incentives for Use/Teaching 53
4.4.3 Incentives for Early Users 53
4.4.4 Incentives for Development of Reusable Software 54
4.4.5 Incentives for Use in Research 55

GLOSSARY 57

PREFACE

In 1975, the Department of Defense (DoD) began the process of standardizing the high order languages (HOL) used to write software for embedded computers. The reason behind this was twofold:

(1) problems associated with HOL design such as:
- creating incompatible dialicts when modifying existing HOL, or
- creating another entirely new languages which was very limited in application

(2) high cost associated with language proliferation such as:
- direct cost of language design and compiler implementation efforts
- language maintenance cost
- cost of schedule slippages and testing problems

The first step was to form a High Order Language Working Group (HOLWG) to identify DoD's requirements for computer programming languages, evaluate the existing languages, and recommend the implementation and control of a "minimal set".

The first requirements analysis, STRAWMAN, was published in 1975 and sent out for review. Based on comments received, a WOODENMAN document was written and again sent out for review. In June 1976, TINMAN was published based on inputs from the services, industry and academia.

Prior to the publication of TINMAN, DoD issued DoD Directive 5000.29 (Management of Computer Resources in Major Defense Systems). The Directive stated that only DoD-approved languages were to be used in Defense System Projects. It required a control agent for each DoD-approved language and established a Management Steering Committee for Embedded Computer Resources (MSC-ECR). On 24 November 1976, DoD Instruction 5000.31 was issued to implement DoDD 5000.29. This "Interim List of DoD Approved High Order Languages" established seven approved HOLs--FORTRAN, COBOL, CMS-2, JOVIAL-J3, JOVIAL-J73, SPL-1, and TACPOL.

Since these HOLs were not considered the long term solution to DoD's programming needs, the Defense Advanced Research Projects Agency (DARPA) was assigned the responsibility of managing the development of a new common HOL. An international request for proposals was issued for the new common language using IRONMAN (derived from comments on TINMAN) as the basis for the competition. Four companies were selected from seventeen responses to develop preliminary designs. In April 1978, designs prepared

by Intermetrics and Honeywell/Cii-Honeywell Bull were selected for further refinement and completion. STEELMAN (Reference II), the final requirements document, served as the standard for this final competition. After wide public review, DoD selected the language developed by Cii-Honeywell. In 1979, the new high order language for DoD was named Ada, in honor of Augusta Ada Byron, the Countess of Lovelace.

On 12 December 1980, the Under Secretary of Defense for Research and Engineering established the Ada Joint Program Office (AJPO) to manage the DoD's effort to implement, introduce and provide life-cycle support for Ada. These objectives were elaborated on in the Ada Program Management Plan (Reference IV):

(1) The AJPO must ensure the implementation and maintenance of Ada as a consistent, unambiguous standard recognized by the DoD and also by the widest possible community.

(2) The AJPO must ensure the smooth introduction and acceptance of Ada in the DoD as early as possible consistent with the needs of individual components.

(3) The AJPO must ensure the provision of life-cycle support for Ada through the development of a robust Ada Programming Support Environment (APSE) to improve productivity both in software system development and in continued system evolution.

It became clear that the Ada Language and the Ada Programming Support Environment were not the only technologies needed to solve the problems identified from 1973 through 1985. Therefore, in July 1982, a joint task force was established to analyze software problems. In early 1983, the Software Technology for Adaptable Reliable Systems (STARS) Program was established to attack mission critical computer software problems. The goal of the STARS Program is to achieve greater systems reliability and adaptability while hopefully improving software productivity, particularly in the post delivery phase which amounts for as much as 80% of the system cost.

The STARS program is a seven year effort in which Ada will serve as a focal point. Areas of common interest between the STARS and Ada programs include:

(1) software reliability/adaptability
(2) software portability
(3) development of software tools
(4) educating the software community

Since its establishment in 1980, the AJPO has been concerned with promoting the adoption of the Ada programming language and its environment. This document outlines past, present and future objectives of the AJPO.

Each Section is discussed using the following format:

(1) Objective: What is the objective behind this section or subsection?

(2) Strategy: What is the reasoning behind this statement--why is it important?

(3) Progress: What has been done to reach this objective?

Section 1.0 discusses the language standardization from its origin through life-cycle support. Section 2.0 covers four two main areas of technology: support environments, methodology, reusable software, and run-time systems. Section 3.0 outlines the procedures to obtain the human resources necessary for widespread use of Ada. The final section concerns Ada business environments; the incentives/strategies used to introduce Ada to the software community.

The thirteen references used in this document are listed below and may be obtained through NTIS or IITRI.

Reference I	Charter
Reference II	Program Management Plan
Reference III	Services' Plans - Air Force - Army - Navy
Reference IV	Validation Policies and Procedures
Reference V	Ada Board Charter
Reference VI	MOU Behind KIT
Reference VII	STEELMAN
Reference VIII	STONEMAN
Reference IX	METHODMAN
Reference X	EDUCATIONMAN
Reference XI	Revised 5000.29
Reference XII	5000.31
Reference XIII	Trademark Guidelines

Comments and questions concerning this document and/or the activities of the AJPO should be directed to:

Ada Joint Program Office
CSS/AJPO
3D139 (400 AN)
Pentagon
Washington, D.C. 20301
(202) 694-0208
AV 224-0208

1.0 LANGUAGE STANDARDIZATION

Objective: Two major goals of the Ada Program are -- one, reusability and portability of software; and two, support of the entire mission critical computer system (MCCS) life-cycle, which is extremely long for many Department of Defense (DoD) systems. A stable and universally recognized language standard is necessary to realize these goals and establish an environment in the software industry in which it is possible for the DoD to take advantage of developments produced elsewhere.

Strategy: A difficulty experienced by users of most MMCS computer languages to date is the failure to control adherence to the definition by implementors. This results in proliferation of dialects through subsetting, supersetting, and inconsistencies. Full advantage of Ada will be realized only if there is a single, widely accepted definition of the language and conforming implementations. Wide acceptance of this standard will reduce the likelihood of incompatible dialects.

1.1 Standard Ada Language

Objective: A standard should be developed which unambiguously defines the Ada language. The AJPO will pursue standardization with the following standardization bodies:

- Military Standard (MIL-STD)
- American National Standards Institute (ANSI)
- Federal Information Processing Standard (FIPS)
- International Organization for Standards (ISO)
- NATO

The goals of these bodies are consistent with those of the DoD.

Strategy: Encouraging standards organizations outside the DoD to adopt Ada will further the acceptance of Ada outside the DoD. This wide standardization also will provide a large base for Ada and protect it from the special interests of any specific group.

1.1.1 Language Reference Manual

Objective: The Language Reference Manual (LRM) is the principal document used to define the Ada language. It provides the basis for initial standardization and configuration management of Ada. This manual must be maintained to:

- clarify uncertainties resulting from English descriptions, and
- rectify inconsistencies between language features.

Strategy: The LRM must be clear, understandable and unambiguous. There will be other documents that seek to explain, formally describe, or illustrate language feature interactions; but these all must conform to the language defined by the LRM, even though they may eventually supersede it as the defining mechanism.

Progress: The Ada July 1980 LRM was the initial language definition. It was designated MIL-STD 1815 on December 10, 1980. The language designer maintained the LRM through October 1982 as part of the ANSI standardization effort. The worldwide distribution of the LRM resulted in detailed questions from Ada language implementors, public and standards review committees, and initial applications programmers. With the completion of ANSI standardization on 17 February 1983, the LRM was frozen for approximately 5 years. This action will stabilize the language during ISO standardization and allow users to gain experience in using the language before language revisions are undertaken.

Until the LRM is revised, language interpretations will be maintained by the Ada Board (see Section 1.2)..

1.1.2 Military Standard

Strategy: A DoD MIL-STD is a contractual requirement when referenced.

Progress: Ada was designated MIL-STD-1815 on December 10, 1980 and then the revised ANSI Ada was adopted on January 22, 1983 as MIL-STD-1815A. When the language is adopted as a standard in a larger community, the MIL-STD will be updated accordingly and reference the more widely accepted standard.

1.1.3 ANSI Standard

Strategy: Ada as an American National Standards Institute approved language will provide wider familiarity with the language because of outside participation in conjunction with ANSI standardization.

Progress: The ANSI canvass procedure was used to obtain the ANSI standard. The first round of the canvass was completed on March 5, 1982. Of the 96 canvassees, only four indicated dissatisfaction with the revised LRM and six chose not to vote. The LRM was then updated and ANSI authorized a supplemental canvass to ensure that the revised LRM was appropriate for standardization. The supplemental canvass began in July 1982, was completed in September 1982 and on 17 February 1983, Ada was approved by the ANSI Board of Standards Review as ANSI/MIL-STD-1815A-1983. ANSI acceptance was a vital step towards international standardization.

1.1.4 FIPS

Strategy: While there is little immediate impact on the DoD for FIPS beyond that provided by a MIL-STD, other agencies

of the U.S. Government will find it more convenient to use Ada once a FIPS is established.

Progress: FIPS are established by the National Bureau of Standards (NBS). In the case of languages, this action usually follows adoption by ANSI. The AJPO is working with NBS on initiating this process.

1.1.5 ISO Standardization Activities

Strategy: ISO standardization is valuable since it applies in those areas where DoD has the least influence and, in most instances, supersedes national standards. It is also the appropriate level of standardization considering the international nature of the computer industry and DoD/NATO and western MMCS systems.

Progress: Considerable technical input has been solicited and received from the international community. The Council of the European Community (CEC) has been closely involved with the Ada program since the beginning and now has a considerable investment in the language and tools for its support. They are particularly interested in its international standardization. Japanese industry has also been involved and set up a group to coordinate Ada activities as have many other nations. The Swedes, Danes, and Japanese have each published the LRM.

The ISO Technical Working Group on real time languages has been involved on a contributing level during the entire Ada development project. Ada was submitted as a work item to ISO TC/97 in 1982. The AJPO was designated convener of a Working Group (WG 14) for final development of an international standard. The Working Group adopted a resolution to encourage ANSI to submit ANSI Standard Ada as the basis for a Formal Semantic Definition (FSD).

The ISO process was suspended pending the resolution of certain questions that had been raised by the TC97 Advisory Group. Those issues were resolved and WG 14 will be formally convened in Paris, France on 10-11 April 1984. The Director of the Ada Joint Program will act as the convener of the ISO Experts Group and invitations have been issued through national standards bodies. A second and third meeting are scheduled for June 1984 in Brussels and November 1984 in Washington, D.C.

1.1.6 NATO

Objective: Although NATO does not establish its own standards, NATO can establish a voluntary agreement for standardization among its members. These standards are generally accepted ISO standards.

Progress: A Working Group (MCCISWG), including a representative from the AJPO, was established by NATO to investigate Ada standardization -- why should NATO recognize Ada, what is the cost involved, and when should this happen. The initial working group's final report was adopted recommending Ada be adopted starting in 1985 if sufficient compilers and tools were available. A second working group was then established to assess Ada's availability and costs for 1984. That working group's final report should be completed in July 1984 and a plan of action for Ada's introduction presented at the Fall 1984 NATO meeting. The plan will include establishing a voluntary agreement for standardization in early 1986.

1.2 ADA Board

Objective: Configuration control of the language, as defined by the LRM, must be implemented.

Strategy: Although the language must remain stable, there must be a formal mechanism to consider, resolve and interpret questions and, if appropriate, issue implementation guidelines. A formal process for approving proposed language interpretations, which is consistent with the requirements of the various standard activities, must be instituted.

Progress: The AJPO originally proposed an Ada Liaison Organization (ALO) in 1981 with responsibilities and functions consistent with configuration control requirements necessary for standardization. Renamed the AJPO Director's Advisory Board (ADA Board) in September 1982, this group's proposed responsibilities include reviewing and recommending to the AJPO positions relative to the administration of Ada-related standards or specifications.

Objective: A means must be provided for resolving questions on interpretation of the language definition, carrying out standards' maintainability activities, and keeping the AJPO abreast of developments in the industrial community.

Strategy: Several different types of questions concerning the language definition occur during its use. All questions from legitimate implementors pertaining to interpretations of the manual must be resolved. More complex questions may call for a full derivation from the axioms of the LRM to assure a consistent implementation strategy. Some questions may arise during the early use which require clarification of the definition. All of these must be addressed promptly and authoritatively. All answers must be circulated immediately and later formalized so that misinterpretations are corrected immediately and not allowed to recur.

Progress: A Charter for the ADA Board (Reference V), along with the appropriate paperwork as required by the Federal

Advisory Committee Act, has been sent to the Office of the Secretary of Defense (OSD) for final approval. Although not required by the statute, the By-Laws of the ADA Board have been developed for internal guidance. The By-Laws outline five major tasks for the Board:

(1) To review and recommend to the AJPO positions relative to the administration of Ada-related Military standardization documents, American National Standard(s), and other related standards or specifications.

(2) To provide advice for official interpretations of the Ada language and of the Standards that are issued by the AJPO.

(3) To provide the AJPO with a source of expertise addressing the Ada language operating environment.

(4) To advise the Director of the AJPO of Ada language, and maintenance issues arising from Ada validation activities.

(5) To advise the Director of the AJPO of current Ada related activities in the industrial and military user environments, and recommend activities the AJPO should undertake to promote the acceptance of Ada.

The AJPO Director will serve as Chairperson for this advisory board.

Issues which can not be resolved by this question/answer process will be referred to the AJPO for resolution within the context of the standardization activity.

1.3 Compiler Conformance

Objective: A compiler vaildation capability consistent with the state-of-the-art must be developed to ensure conformance of Ada language translators to the Ada standard.

Strategy: The benefits for program and tool portability can not be realized unless every compiler conforms to the standard. Previous practice in the industry has been for the compiler to be a de facto definition for the language. Consequently, when compiler developers make different implementation decisions, dialects appear.

Progress: The Ada Validation Organization (AVO) has been established to ensure that Ada compilers implement the same common language. The Ada Compiler Validation Capability (ACVC) will serve as an enforcement tool to regulate conformance of compilers to the Ada standard.

1.3.1 Ada Validation Organization

Objective: An Ada Validation Organization (AVO) has been established to maintain the validation suite and to ensure all Ada compiler validations are conducted in the same manner so that the results are equivalantly acceptable.

Strategy: Since there is a firm Ada standard, it should be possible to enforce the standard by requiring all compilers to be checked for conformance with the standard. Without a powerful compiler validation capability, assurance that a compiler implements the Ada language correctly is not possible. Even with the best of intentions, there is virtually no chance that two compilers will implement precisely the same language without some mechanical check. The existence of a validation

capability will not only assure conformance, but will be of enormous benefit to implementors and provide a considerable measure of confidence in the correctness of the implementation. It is vital that all implementations be validated. Otherwise, there will be dialects and the benefits of sharing software within the entire community will be lost.

This AVO function will remain within DoD to ensure that positive control of implementations of the standard is maintained. However, satellite validation organizations will be used to conduct validation of compilers both inside and outside DoD's area of interest. The AJPO is in the process of establishing three satellite organizations with additional ones being established when desirable.

Progress: A draft set of policies and procedures (Reference IV) for the AVO was distributed for public comment and will be used to conduct the AVO through 1984. A revised policies and procedures document will then be developed based on the lessons learned in the first two years; will be publically reviewed; and then will be used until the language is revised or established as an ISO standard.

1.3.2 Ada Compiler Validation Capability

Objective: The AVO will maintain the Ada Compiler Validation Capability (ACVC), will disperse copies of the test suite and conduct validations and monitor satellites to ensure that validations are equivalent for the AJPO.

Progress: A test suite of Ada programs, and an Implementor's Guide to define test objectives and discuss the ramifications of critical sections of the LRM have been developed. Approximately 1400 test objectives were documented in the Implementor's Guide and a set of over 1400 test programs

were upgraded to reflect the language changes made during ANSI standardization.

Additions to the test suite will continue on a semi-annual basis with the aim of improving the state-of-the-art in compiler validation to further ensure conformance to the standard. Since other verification and validation (V & V) activities will exist for Ada implementation, and as practical, these V & V capabilities will be used to augment the ACVC efforts.

To date there have been four ACVC releases:

(1) 1.0 - 18 February 1983
(2) 1.1 - 11 March 1983
(3) 1.2 - 1 August 1983
(4) 1.3 - 1 December 1983

and three successful compiler validations (11 April 1983, 13 June 1983, and 9 August 1983). Versions 1.4 and 1.5 will be released 9 April 1984 and 10 December 1984, respectively.

1.4 Activities Clarifying the Language Standard

Objective: As part of the Ada programming language standardization effort, steps must be taken to improve implementations of the language, establish a formal language definition, evaluate language change proposals and prepare for the replacement of Ada by a more advanced language.

Strategy: Even though the Ada programming language has been standardized, there will be a need for interpretations and minor clarifications. While these functions of maintaining the existing language fit into earlier objectives related to clarifying the language definition, as Ada is used, possible improvements will be noticed.

Language change proposals need to be carefully evaluated in terms of programming methodologies, consistency with the rest of the language, and configuration control of environments and applications. Some of these controls will be managed through the AVO as part of the existing language standardization, others will be define. These should be coordinated through the AJPO as part of a careful evaluation of long term improvements to the language.

1.4.1 Implementor's Guide

Objective: Although the LRM defines the language, there are numerous questions on how certain features relate to previous languages, machine hardware features,and operating systems. These issues are important during compiler and run-time design and implementation.

Strategy: Previous practice in the industry has been for language implementors to make different decisions based on their own goals rather than completely following a standard language interpretation. It is important that there be a single source for information and decisions on these types of issues. In addition to answering specific questions, an Implementors' Guide should also point out trouble spots which might otherwise be missed and lessons learned to assist early and new implementations.

Progress: A contract was awarded based on a competitive procurement in September 1979 to develop the Implementor's Guide in conjunction with the ACVC development. Preliminary versions of the Implementor's Guide were produced for the 1980 version of the language definition. The guide is being revised to comply with the February 1983 ANSI/MIL-STD-1815A version of the Ada language and will be available during the fall of 1984.

1.4.2 Operational Semantic Definition

Objective: It is important to develop an implementation of Ada to illustrate the semantics of the Ada language, its implementability and to test the ACVC capability.

Strategy: An early implementation serves as an important check on the consistency of the Ada semantics, an operational model for those implementing the language and an independent check on those implementing the ACVC.

Progress: New York University (NYU) completed an Ada interpreter for the initial language to prove Ada's implementability. The NYU translator was updated to reflect the ANSI changes to the language and on 11 April 1983 was the first Ada language translator validated by the Ada Validation Office (AVO). An independent version of this translator which runs faster and uses error diagnostics is being supported by the Army to be used as an educational tool.

1.4.3 Formal Semantic Definition

Objective: A formal semantic definition of the Ada language should be developed.

Strategy: Although the LRM is the current definition of the language, it is based on English descriptions. A formal semantic definition will provide a more precise way to describe the meaning of the language constructs. Although it appears that there is an insufficient theoretical foundation for the formal description of some Ada contructs such as parallelism, it is desirable that the language be as precisely defined as the state-of-the-art will permit.

Progress: In December 1980, the French Government Laboratory for Information Processing (INRIA), under subcontract

to the language designer, developed a Formal Semantic Definition (FSD) of the 1980 version of the Ada language using denotational semantics. The University of Southern California (USC) Information Sciences Institute attempted validation of that definition.

Based on the results of that validation and public comment and changes in the language based on the ANSI standardization process, the denotational semantic definition was to have been revised. However, criticism of the initial definition by the ISO Experts Group has required re-evaluation of the FSD approach. Based on the concerns expressed, an international team of researchers has been asked to investigate the alternative formal semantic models and to recommend an approach. A formal semantic definition may be developed as part of the ISO standardization process.

1.4.4 Replacement of Ada

Objective: As the Ada language is used and as the technology used for language design matures, the Ada standard will need to be updated and possibly even superceded.

Strategy: Although Ada ANSI/MIL-STD-1815A-1983 is the language that the DoD has adopted at this time for embedded computer applications, it is certainly not the ultimate answer to DoD software problems. Thus, DoD must continue research into more advanced language features and prepare for the revision and ultimate replacement of Ada when that becomes the optimum strategy. Although it may be possible (and even desirable) to incorporate many new features into Ada to enhance its utility the next decade or two, the advantage of extension must be balanced against the need for consistency and stability.

Changes to the Ada standard must be adopted only after careful analysis and in such a way that they are consistent with the Ada design and the language standardization process. When the state-of-the-art in language design progresses to the point where a new language is the appropriate choice, the AJPO will provide for such development and devise a strategy for a smooth transition.

Progress: It is anticipated that Ada will have a useful DoD life cycle of at least twenty years. The AJPO continues to monitor language developments in the research community. Although it does not conduct or sponsor such research, it will encourage extensive experimentaton with Ada and investigation of possible extensions to ensure a thorough understanding of the capabilities and a rational approach to its evolution. The AJPO will provide analysis of candidate features for future extensions to Ada. No specific plan is appropriate at this time for either the extension or replacement of Ada. If the various Ada standardization bodies do not sponsor a comprehensive review of Ada at least every five years to determine if any action is appropriate at that time, the AJPO will do so.

2.0 TECHNOLOGY

Objective: The purpose of this section is to describe the mechanisms necessary to make maximum available use of an effective and efficient High Order Language (HOL) through the use of a sophisticated programming support environment complete with advanced development, maintenance and management tools.

Strategy: The Ada program is much more than just a language standardization effort. It includes controlling the cost and improving the quality of the software by facilitating the application of modern software engineering practices to mission critical computer system (MCCS) developments. By design, Ada incorporates many of the features needed to support modern programming practices; but, just like any tool, it can be misused. This objective will provide Ada MCCS applications with reliable, modern, automated tools for support throughout their life-cycle. It includes the design, development and distribution of complete life-cycle environments, along with easily transportable tools. The capabilities of Ada will be fully realized only when a sophisticated Ada Programming Support Environment (APSE), complete with advanced development and management tools, is made available and widely used.

MMCS computer systems have to contend with many difficult life-cycle software problems. They are often very large (sometimes in the millions of lines of code), usually long lived (lasting more than 20 years), and usually require continuous changes (typically several times a year) in order to keep up with new weapon systems or change in threats. In recognition of these characteristics of the MMCS computer application area, the High Order Language Working Group (HOLWG) included support environments as a critical part of the U.S. DoD's HOL

standardization effort. An integrated software environment containing a good set of tools would also encourage acceptance of the language, thereby magnifying the benefits of the language standardization effort. They felt that an integrated software environment containing a good set of tools would encourage acceptance of the language, thereby magnifying the benefits of the language standardization effort.

The set of tools required for MMCS system life-cycle support environment were not only the traditional editors, compilers and other traditional program development and documentation aids; but also tools supporting the complete weapon system program life-cycle, including the often ignored management process. Failure to develop such a facility would mean that software development would most likely continue to be treated as an art, with little improvement in a program manager's ability to predict software costs and completion times; and little reduction in the number of late, erroneous and costly software.

The objective of supporting MCCS Ada users with automated tools has four main sub-objectives:

- Life-cycle Environments
- Methodology
- Reusable Software
- Run-time Environment

Development of support environments for MCCS applications; development of system life-cycle methodologies, development of reusable software; and reusable runtime software are discussed below. The fourth, will be discussed in Section 2.3.

2.1 Support Environments

Objective: Software Life-cycle Support environments for MCCS applications need to be developed.

Strategy: Although Ada does not require a special environment, the life-cycle maintenance of MCCS systems does.

2.1.1 Ada Programming Support Environment Requirements

Objective: The establishment of the requirements which APSEs must support over the entire software life-cycle of MCCS programs must be established.

Strategy: Without an appropriate understanding and documentation of the requirements, we will not know if the APSE's which will be constructed, are complete or not. A robust APSE, complete with advanced development and management tools, will provide the opportunities for substantial improvement in life-cycle software management. Although each Service employs a different strategy in the acquisition and management of software, a set of tools which can be shared should be developed. The AJPO will develop or carefully coordinate development of tools common to the need of all services and coordinate development of service specific tools.

Progress: The initial DoD-sponsored environment workshop was held in June 1978 at Irvine, California, to discuss alternatives. The result of the workshop was a draft requirements document, "PEBBLEMAN," describing all aspects of the problem, including many policy issues. In November 1979, a revision was published treating only the technical issues and a workshop was held at San Diego, California, for review and solicitation of new ideas. The final "STONEMAN" document, defining the high

level requirements for an APSE, was prepared and distributed in February 1980 (Reference).

The STONEMAN requirements for the APSE were used by two DoD contractors and several U.S. and European groups in the design for their environments. As experience was gained with STONEMAN APSE's requirements, and as APSE requirements became better understood, a refinement of the "STONEMAN" document became necessary. All such revisions or follow-on documents will be widely circulated for comment. There are already several recommended clarifications which are being consolidated and documented. A draft of the document was available the 2nd quarter FY 83. A follow-on document may be circulated in FY 84. This task has been assigned to the Navy (NOSC) as part of the Kernel Ada Programming Support Environment Interface Team (KIT) effort.

2.1.2 Tool Portability

Objective: As APSEs are developed, tool portability should be established.

Strategy: A major sub-objective of the Ada Program and the STONEMAN model is to amortize the cost of very expensive life-cycle tools by installing them on as many DoD APSE's as possible. The STONEMAN assumes that environment portability can be achieved by re-hosting an inner layer of software (Kernel Ada Programming Support Environment-KAPSE) which provides a virtual interface between the hardware with its operating system and the rest of the APSE. While this would really be a re-implementation of the KAPSE on the new host, the APSE toolset would be moved without requiring any re-programming.

The STONEMAN model of portability is correct if only one KAPSE is implemented on all hosts. Unfortunately, since the

concept of an APSE is relatively new, it was decided to develop two DoD KAPSEs. In addition, there are at least two European efforts underway and several U.S. and European commercial KAPSEs. The AJPO anticipates that industry, academia and other governments will support and build their own APSEs. Based on this concept, the STONEMAN mechanism for achieving portability was recognized as inappropriate. An equivalent mechanism to rehost the standard KAPSE could be achieved if all KAPSEs implemented the same set of KAPSE-to-tool interfaces. This would permit experimentation with a wide variety of KAPSE implementations while still achieving the DoD's goal of APSE portability.

To this end the AJPO drafted and had the Deputy Under Secretary of Defense for Research and Engineering (Acquisition Management) and the three Service Assistant Secretaries for Research and Development sign a Memorandum of Agreement (MOA). The operating procedures for the MOA provided for the Navy to lead a joint service KAPSE Interface Team (KIT) to identify and recommend conventions for DoD supported APSEs for tools-to-KAPSE interfaces.

Progress: The Navy KIT and an Industry/Academia support team (KITIA) accomplishments include:

- Establishment of basic definitions
- Documentation for interface categories
- Documentation for KIT APSE I&T implementation strategy
- Drafted statement of requirements and criteria
- Held a configuration management workshop
- Published three public reports (April 1982, October 1982, June 1983)

- Draft MIL-STD CAIS
- Draft requirements and criteria document

The degree of success of this effort at identifying reasonable conventions and standards and the degree to which all KAPSE developers adhere to them will significantly influence the cost of porting a tool from one KAPSE to another. It will also influence how well we can amortize the cost of sophisticated tool development across a large number of installations.

The desirability of a single KAPSE design must be tempered by the relative risk since the effects of design decisions in this area are not well understood. While the conceptual goal is to develop a single APSE, it is prudent at this point to support parallel developments while working toward the definition of conventions to permit portability of tools and programmers between APSE's.

In light of the complexity of the problem, the concern that we not over constrain the future APSE developments and the immediate short term requirement for tool portability between DoD APSEs, a three phase approach to KAPSE interface standards is being proposed.

In the short term the AJPO, Navy and two DoD APSE developers are investigating ways of reducing the differences between the KAPSE interfaces found in the Ada Language System (ALS) and the Ada Integrated Environment (AIE). This investigation will provide DoD tool correction and some conventions for other KAPSE implementors to follow. In the mid-term a detailed set of requirements for Common APSE interface standards (CAIS) will be developed and implemented. Public review of the draft specification of the CAIS was held 14-15 September 1983 and a second review will be held in August 1984

with CAIS Version 1 being submitted as a mil-std in January 1985. Version 2.0 should be initiated in January 1985 and submitted for public review and standardization in 1987.

2.1.3 Conventions and Standards

Objective: Conventions and standards must be developed to support the environments.

Strategy: Although the requirements for an APSE have been defined in STONEMAN, its design implies the definition of conventions and specifications for the interface between tools, users and data bases. These conventions and specifications permit the consistent introduction of new tools into the environments, the movement of programmers between APSEs and the movement/sharing of projects between APSEs.

2.1.3.1 Definitions

Objective: The first sub-objective in this development is to ensure that conventions and standards are defined for interfaces to the KAPSE so that independently developed tools can interact. Two critical aspects of this objective are to ensure interaction and that interfaces to the KAPSE are sufficiently general, yet well defined, to permit independent development of advanced tools.

Strategy: The existence of conventions, standards and interface specifications permits independent development of tools that can interact in a useful manner. There is virtually no hope of meaningful portability of tools without the definition of such standards. This objective must be approached carefully to avoid a premature definition of such standards. A part of this sub-objective is the identification of appropriate validation mechanisms for KAPSE standards.

Progress: The Navy has initiated the KIT activities to define the interim short-term interface conventions under development by SofTech and Intermetrics as part of the MOA initiative. When appropriate conventions are identified, they will be submitted to the AJPO for publication as MIL-STD requirements and insertions for other appropriate documents.

Other areas concerning definition of conventions will evolve, particularly in the tool-to-tool data interface area. Initial steps are being taken in cooperation with the German Ministry of Defence (MOD) and contractors for various ongoing implementations to establish conventions and possible standards. Contact has been made with the ANSI Job Control Language (JCL) committee to ensure compatibility with its standard activity. The AJPO will actively monitor environment developments to ensure the definition of conventions and standards. These conventions will be used to evaluate the concepts before standards are sought.

There are many tool-to-tool data interfaces which may ultimately be standardized, such as those used between various text processing tools, compiler-oriented tools and tools used for requirements and management support. A coalition of the TCOL and AIDA proponents, in cooperation with the German MOD have developed a Descriptive Intermediate Attributed Notation for Ada (DIANA). The AJPO currently encourages the use of DIANA as a convention and has a contract with Tartan Laboratories to: (1) maintain configuration control of that representation, (2) make appropriate changes to DIANA to reflect ANSI Ada and its production use, (3) provide technical answers to questions raised as a result of using DIANA and (4) manage the general acceptance of the revised manual.

APSEs will undoubtedly be developed independently by academia and industry. These APSEs will not all be compatible with the chosen conventions; however, tools which are sufficiently powerful may be modified to interface with the DoD APSE. The AJPO will foster the development of a complete and powerful APSE so that it will become the leading candidate to evolve as the predominant support system. This will encourage designers of independently developed tools to conform to the DoD chosen conventions.

One area which is not being addressed currently, but which needs to be, is standardization of some minimal part of the user interface. This is important if users are to become portable between different APSEs with different tool sets. It is important that the mechanics of using tools be similar even if the tools behave differently, and the rationale for using different tools be the same for some minimal set of functions.

2.1.3.2 Tool Development

Objective: Tools must be developed to ensure consistency with DoD APSE requirements.

Strategy: Although tools will be developed in the marketplace, the AJPO must make sure that DoD tools are consistent with DoD APSE requirements and other components of the APSE.

Progress: The AJPO is sponsoring NBS's definition of functional requirements for a complete taxonomy of tools. This taxonomy will be given wide public distribution to solicit comments and then coordinated with the services and agencies to develop a prioritized list of tools to be developed or purchased for use in the DoD APSE.

Development of specific tools will be undertaken when the design specification for the KAPSE interface is completed. This

activity will begin in parallel with the implementation of APSEs. The AJPO will work closely with the services to identify unique tool requirements. Special attention will be devoted to the definition of suitable interfaces and intermediate representations. The AJPO will coordinate with the MSC-ECR R&D panel to encourage submission of industry internal research and development plans for evaluation so that AJPO activities can take advantage of industry developments.

Progress: The Army has a MAPSE development contract with SofTech Inc. based on a competitive procurement to develop the ALS which will be hosted on the VAX-11/780 (VMS) and targeted to the VAX-1/780 (VMS). They began testing in January 1983 and plan to have a validated Ada compiler in October 1984. A MCF code generator is scheduled for release in October 1985.

The Air Force has a contract with Intermetrics to implement the AIE which will be hosted on an IBM 370. Code generators will be developed for MIL-STD 1750 and other potential embedded machines. Early 1985 is the target date for compiler validation with a full system scheduled for release in 1985.

The Navy will rehost and retarget an APSE based on the ALS. The AJPO will ensure that areas of possible standardization are clearly identified and coordinated.

A consortium of European industry has a similar effort underway with support from the CEC. The AJPO is coordinating with the CEC so that by sharing technical information, relative risk is reduced. A standardized program development system (SPERBER) which is based on Ada and the STONEMAN requirements is being developed by the Germany MOD. A MOA is being pursued by the AJPO to permit the sharing of technology and software between the German MOD and U.S. DoD.

The availability of a high quality Ada tool in an integrated environment may decide the ultimate acceptance of the Ada language. Sharing of tools will prevent duplication, ensure better support tools, and generally encourage use of the language. To the extent possible, Ada tools developed by DoD or under DoD contract will be written in Ada and made publicly available. If Ada is not to be the tool implementation language, the AJPO will provide a written waiver to that effect. The decision to use another language for any DoD-sponsored APSE tools will be made by the AJPO in response to a written request and AJPO evaluation of the rationale and supporting analysis provided with the request.

Progress: The details of implementing a tool repository have not been developed. There are a number of technical and procedural difficulties which must be studied. It is unlikely that such a facility will be useful before FY 84. Earlier tools will be distributed as available on a case by case basis. Government developed tools, subject to appropriate export controls, will be placed in the public domain where they will be available to private software firms for integration into their own software products. The AJPO will seek a general or limited license for export of software tools generated by DoD and obtain the rights for those produced by other governments.

The AJPO will seek to maintain an exemplary and evolving environment complete with advanced tools which may be applied experimentally. Although the sponsor of specific tool developments may be responsible for its maintenance, the AJPO will manage distribution via the repository. A configuration management plan must be developed to support this repository.

2.1.4 E&V Effort

Objective: Acceptance tests are insufficient as the sole means for assessing and selecting APSEs due to the differing properties of both APSE implementations and applications.

Strategy: Steps must be taken to develop the technology which will be used as the basis for the evaluation and validation of APSEs. This technology should consist of both the techniques and tools necessary for performing assessments of the APSEs and the conformance of the APSEs to existing or future established standards.

Progress: An Evaluation and Validation (E&V) effort has begun to develop this technology. Publications will inlcude an E&V quarterly status report, an annual public report and annual workshops. Working groups will be established to analyze:

- requirements
- technical coordination
- public coordination
- the CAIS validation capability
- APSE validation procedures

As the need arises, other working groups may be established.

2.2 Methodology

Objective: Consistent system life-cycle methodology which can capitalize on the software engineering structure and propose specific automated tools to support that methodology needs to be developed.

Strategy: The AJPO must actively seek to develop an Ada and system development culture which can support the promise

that Ada offers. The AJPO must develop, encourage and foster those facilities that will make the use of Ada more productive. Consistent with the goal of making Ada the language of choice, the AJPO must embrace the philosophy of adopting and demonstrating exemplary tools which will naturally be chosen by informed managers. This objective will be pursued following three steps which will be somewhat iterative in nature.

2.2.1 Requirements

Objective: The requirements for a life-cycle methodology which can support Ada MCCS systems must be established.

Strategy: This methodology must be cognizant of hardware/software tradeoffs and capable of deferring such decisions as long as possible. This will provide for insertions of future hardware/software technologies and later substitutions of one for the other over the life of the system.

Progress: The UK Department of Defense Instruction (DoDI) Development Methodology Study assessed Ada as an acceptable HOL for implementing software developed under a wide variety of methods. The properties of such a software development methodology were initially defined in a document called METHODMAN by Wasserman and Freeman (Reference IV). The first draft was issued for limited comment in July 1982 and a second draft was more widely distributed in October. Based on the comments received from these drafts, the Ada Methodologies Project was established in Spetember 1983. This project is concerned with the full sprectrum of software life-cycle methodologies that capitalize on the support provided by Ada for the use of modern software engineering methods. The Methodologies Coordination Team (MCT), with representatives from each service and technical consultants from industry, will oversee this project.

2.2.2 Life-Cycle Methodologies

Objective: After the requirements are fairly well established, the AJPO must encourage the development of one or more

methodologies to meet the requirements. This must include an evaluation of existing methodologies.

Progress: There is currently no coordinated review of existing methodologies, although it is obvious that each APSE tool set will support a certain methodology. It is anticipated that this will not be a focal point until the Software Technology Initiative (STI) takes over the task.

2.2.3 DoD Standards

Objective: After a sufficient evaluation and review of the various methodologies, the AJPO must recommend DoD standards for life-cycle methodologies. In the mean time, all proposed standards must be reviewed to ensure that they do not prevent the future development of MCCS methodologies and at least permit the use of modern software engineering practices as we understand them today. The AJPO will always be responsible for this aspect of methodologies since it is an implementation problem.

Progress: There are currently two proposed standards which impact MCCS methodologies: MIL-STD-1679A and MIL-STD-SDS. The intention was to have 1679A be the near term standard and SDS incorporate the long term standard engineering practices as they are proposed and accepted.

Both documents have undergone a fairly wide review and comments are being analyzed to be included in the final document. It appears that 1679A will be accepted, but SDS will require another major review before it will be an acceptable long term document.

2.3 Reusable Software

Objective: Reusable software needs to be developed to support MCCS Ada users.

Strategy: The Ada language provides for access to function libraries to support the application programmer. Examples of such functions are trigonometric functions, graphics packages, data base management functions, families of operating systems, navigation functions, etc. These libraries would be available to all APSE users.

2.3.1 Standard Packages/Extensions

Objective: There is a wide variety of communities with individualized needs for specialized libraries. Independent development of such libraries can lead to unnecessary duplication and lack of generality. The development of two types of libraries is discussed below.

2.3.1.1 General Purpose Library

Very little is known about obtaining and developing software. Questions like the following need to be answered.

- How do I find a package which satisfies my needs?
- How do I state my requirements?
- How do I describe my package so it can be found?
- What conventions must I follow when constructing my package?

The development of such a library mechanism and appropriate standards will probably start with some initial theoretical studies and experimentation. After the requirements are better understood, an initial draft requirements document and successions should be developed leading to a "CATMAN." Finally, a multi-stage design and development effort will be undertaken.

Very little has been done in this area. Several talks with universities and research institutes have led to several draft proposals but nothing is underway at this time.

2.3.1.2 Ada Mathematics Library

The definition of an Ada Mathematics Library has been drafted and will be implemented in the near future. The requirements for other common libraries will be defined and developed in a similar manner based on the experience gained with the mathematical library.

The AJPO will coordinate the development of a requirements document for applications' libraries and will circulate it for public comment. A library or set of libraries will be developed based on that requirements definition. Independent efforts will be coordinated to avoid duplications. These activities will not be initiated until the Mathematical Library experience is understood.

2.3.2 Reusable Software Distribution

Objective: Steps must be taken to distribute this reusable software.

Strategy: The software industry has long anticipated the time when parameterized software modules may be structured and shared among software modules, or chips may be used in the design of equipment. This objective will attempt to make such a strategy a reality for Ada-based software. Computer network and data base management technology will be used as far as practical and economically advantageous in fulfilling this objective. There are a variety of technical and practical issues to be solved before such a capability is realized.

2.3.2.1 Develop Requirements

The first aspect of this objective will be to develop the requirements for reusable software distribution. This requirements document must identify such considerations as configuration management policies and procedures, maintenance

responsibility, potential user export control, rights and potential threats to the system. Such requirements are effectively external to the Ada Program and must be satisfied by the distribution mechanisms. The relationship to commercial facilities must be established.

2.3.2.2 Distribution Facilities

The second sub-objective is to establish one or more DoD and commercial distribution facilities to handle the distribution of tools and DoD software components. This must include not only the distribution mechanism, but also mechanisms for capturing, documenting, classifying, and cataloguing the software; querying the information to obtain candidate packages; distributing the software with appropriate export control; documentation; test data; design criteria and all other appropriate information.

2.3.2.3 Facility Operation

The final sub-objective will be to operate the facility. This must include software, documentation, catalogue maintenance, configuration management facility operations, publication, announcements, etc. Such a facility must be able to distribute not only the APSE tools but also application oriented packages.

2.3.3 Tools

Objective: The AJPO must support the construction of at least one complete set of tools supporting the methodology of choice.

Strategy: The determination of which tools along with a definition of their functional requirements must be established. Such a tool list will become a part of the overall tool

development strategy. This task will ultimately become part of the STI.

Progress: No coordinated DoD effort is being aimed specifically at the methodology aspect of tool development.

2.4 Run-Time Environments

Once the various application areas supported by standard packages and their requirements are understood, the AJPO will coordinate the development of libraries such as those described above. One application area currently being investigated is the Ada run time system. Since there are no existing Ada Run-Time Operating Systems (ARTOS) for military computers, the development of an ARTOS is required for all initial Ada users. Advanced planning, careful analysis and standard designs must be imposed so that each development has the opportunity to reuse items developed for a different ARTOS. The Ada Program Plan offers the unique opportunity to reduce this redundant and wasteful effort.

The AJPO is sponsoring the development of an ARTOS research and development plan for base target machines. The intent is to initially establish the ARTOS requirements as a family of operating services in a document called "Requirements for Ada Run-Time Operating System Environment." This document will then be coordinated publicly in order to obtain wide participation and a good structure to meet the widest range of military requirements. These requirements will include both application areas and architecture specific considerations.

3.0 HUMAN RESOURCES

Objective: An education and training program must be developed to provide the human resources necessary for widespread effective use of modern software methodologies through Ada.

Strategy: Ada's potential demands a vigorous and visionary education and training program. Through the DoD Common Language Effort, the international computing community has created a powerful and sophisticated tool. Steps must now be taken to ensure the use of Ada by an equally sophisticated computing community.

Over the last twenty five years software technology has made enormous strides. The Ada program serves as a focal point for the consolidation of state-of-the-art thinking about this technology. Proper use of the technology can contribute significantly to controlling the cost and improving the quality of software. However, this is a sophisticated technology, and its proper use depends upon sophisticated education and training for the people applying it. The effectiveness of Ada will be determined by the degree to which people can use it to implement software engineering practices in applications programming. A carefully planned education and training program which teaches both fundamental software engineering concepts and the effective use of Ada is therefore essential.

An effective education and training program can be achieved only through a cooperative effort among the government, industry, and academia. The AJPO is in a unique position to provide orchestration for that cooperation, and this section describes the efforts to be undertaken to achieve that goal.

Progress: The AJPO is in the process of preparing an Ada Software Engineering Education and Training Plan. This plan will outline the scope of the AJPO's involvement in education and training and the tasks to be undertaken by the services. The plan will be subject to public review by interested individuals in industry, academia, and professional societies. Also, the AJPO plans to form an Ada Software Engineering Education Working Group (Ada SEEDWG) to implement, monitor, and modify the AJPO Ada Software Engineering Education and Training Plan.

3.1 Strategic Planning for Education and Training

Objective: An education and training program should be developed to meet the needs of DoD personnel.

Strategy: The Ada education and training program offers an unparalleled opportunity to raise the sophistication of the community in the use of modern software development practices. While these practices have been understood in some communities for many years and are being used by a growing number of practitioners, education is still a vital factor.

Progress: The approach the AJPO will use to provide this capability are outlined in the Education and Training Plan under Strategic Planning. These include:

- Develop an Ada Information Course to answer the basic question: "What is Ada and why should I be interested in it?"
- Develop Ada Software Engineering Familiarization Course/s to change the learner's mind set acquired from previous language experience(s).

- Conduct Ada Study Projects to obtain measurable data concerning instructional effectiveness.
- Develop an Ada education and training technology insertion program and establish a shared data base of Ada education and training experiences.

3.2 Human Resource Management

Objective: Steps must be taken to identify the needs of each community (DoD, industry and academia) to ensure the availability of well-trained Ada personnel.

Strategy: If Ada is to become the HOL language of the software community, it is necessary to provide the human resources to support it. Interaction with each community will provide such pertinent information as:

- Force management--number and types of personnel to be trained
- Career incentives--policy concerning acquisition and retention of personnel; career paths
- Development plans--type of training needed for personnel
- Management on the job--type of manager personnel will encounter when they return to the work environment; feedback mechanisms available from managers on job performance after training

Progress: This is not a task for the AJPO. As outlined in the proposed Ada Education and Training Plan, this operational planning will be done by the individual agencies and monitored by the AJPO. Once the different job categories and skills are understood, the AJPO (or the Ada SEEDWG) will be in the position to recommend the necessary courses, materials, etc. to fit

particular environments. Each Operational Plan will be developed by the defense and government agencies needing them and will define such things as personnel to be trained, schedule guidelines, training facilities and types of delivery methods.

3.3 Information Dissemination

Objective: The AJPO must be aware of available materials that will improve the efficiency, uniformity, and quality of education and training materials.

Strategy: Courses in Ada and software engineering are being offered by industry, academia, and the government. These courses range from one day in-house lecture seminars to five day hands-on workshops to one semester undergraduate courses. Awareness of these activities will allow the AJPO to publish documents articulating the elements, requirements and resources available to DoD, industry and academia. Public exposure and comment to these materials may ultimately lead to refinement and/or development of more effective training materials and trained personnel.

Progress: The Ada Information Clearinghouse (Ada IC), operated by the ITT Research Institute for the AJPO, has prepared an IC Newsletter and the Catalog of Resources for Education in Ada and Software Engineering (CREASE). The newsletter is published bimonthly and provides updates on all AJPO activities. CREASE disseminates information on courses, seminars, training programs, textbooks, etc. which provide Ada and software engineering education and training. The first edition was released in July 1983 and the AJPO plans to update CREASE quarterly.

Other DoD efforts to develop and/or disseminate education and training materials include: participating in and distribution of video tapes, organizing workshops, and the development of an Ada Technology Center at Jersey City State College.

The Army has begun developing an Ada Training Curriculum which defines three types of training modules: Ada Programming Support Environment, Ada Language, and Methodology courses. Fifteen of the 35 proposed courses have been developed and are being taught at Fort Monmouth. Similar efforts are being conducted by the Air Force at Keesler Air Force Base.

Part of the AJPO's responsibility will be to monitor these and other projects to ensure quality training and avoid duplication of available materials and courses.

3.4 Professional Community Development

Objective: Interaction within the community is essential to the life-cycle support of Ada.

Strategy: The Ada education and training program is an ambitious effort to improve the performance of the computing community. Its success will be greatly enhanced by continuous review and commment by all interested communities.

Examples of procedures that may be used to monitor and enhance the Ada education and training program are listed below.

- Collect and consider comments generated throughout the education and training program and recommend improvements.
- Cultivate the community of educators and the community users, including universities, research communities,

and professional societies, to attract their attention to the Ada education and training program.

- Form a working group with representatives from DoD, academia, and industry, and use it to evaluate education and training opportunities and tasks
- Ensure the existence of communication mechanisms, including electronic mail systems (e,g., ARPANET and CSNET), an information clearing house, conferences, and publications.
- Conduct studies to evaluate effectiveness.
- Exploit new technology.
- Build case study files.

Progress: A Software Engineering Education Working Group (SEEDWG) will be established to monitor, adapt, and expand the education and training plan and monitor the education task. This working group will be established initially as a DoD Tri-Service Working Group. Plans are to expand the working group in late 1984 to include representatives from industry and academia. This new working group will be patterned after other AJPO efforts such as the CAIS, KIT/KITIA and E&V.

An Ada-Information account has been established on ECLB for people with ARPANET or Telenet accounts. This data base provides information on various topics including: course outlines, upcoming seminars/conferences, available textbooks, and how to obtain copies of the reference manual.

Other efforts to monitor and enhance this program as outlined above will be undertaken by the AJPO or the SEEDWG during FY 85.

4.0 ADA BUSINESS ENVIRONMENT

Objective: Smooth introduction and acceptance of Ada by the DoD embedded software community and other non-DoD communities is a major management responsibility of the Ada Program. Since Ada and its support environment have the potential to improve the quality of embedded software and control life-cycle costs, the introduction of Ada into the software acquisition process has the potential to save these communities millions of dollars.

Strategy: The momentum of the Ada program has produced a climate ripe for early acceptance; however, the infrastructure for the insertion and adoption of Ada and its support environment must be in place prior to the widespread use of Ada. Ada and its support environment provide techniques for improving the DoD software acquisition process, but misuse of these techniques through ignorance and lack of training could result in more problems -- not less.

In order to ensure the successful introduction of Ada and fielding of Ada implementations, the DoD must ensure:

- that adequate training programs exist
- current practices and procedures are understood and modifications are made to existing methods when needed,
- Ada and its support environment to be adequately tested and training provided,
- the support environment must contain a tool set sufficient for application software life cycle support, and
- the establishment of a maintenance/configuration control organization for Ada software.

Accordingly, a major subtask of establishing an Ada business environment is the creation of introduction strategies which will plan for the development of an Ada support infrastructure containing all these elements. Existing DoD policies will be modified and, where appropriate, new policies issued which will provide procedures for the introduction and use of Ada and modern software engineering.

Once the support infrastructure has been established, means must be provided which make the use of Ada and its support environment convenient and advantageous to DoD embedded computer systems program managers. Some of the mechanisms for facilitating the use of Ada are:

- making the software community aware of the Ada program and the benefits it offers to their programs (introduction),
- promoting program managers' awareness and providing easy access to training programs (implementation) and,
- providing interaction among members of the Ada community through affinity groups (life cycle support).

This effort will be extended to the research environment since facilitating the use of Ada in research will speed the transition of products from research to engineering development, thereby lowering the cost and reducing development time for new products.

After the support structure and facilitators have become operational, steps must be taken to provide incentives to overcome the resistance to change. The use of Ada will produce lower life-cycle software costs; however, like any innovation, the introduction of Ada requires users to embark on an education process. This education process will impose a learning curve

which will initially result in higher costs, schedule impacts, and additional management problems on software developments. Incentives must be provided to program managers who will be early users of Ada to ensure their cooperation and limit the risk of early Ada use.

4.1 Business Environment Strategies

Objective: Developing a business environment strategy is critical to the success of the Ada program. Premature introduction may result in technical difficulties which would, in turn, project a poor image of the Ada program in the software community. Ada should be introduced as early as feasible in order to gain the benefits from Ada as soon as possible. However, this introduction should be very deliberate and well planned in order to avoid as many initial problems as possible.

4.1.1 Develop Organization Strategies

Objective: The first step in developing business environment strategies is the development of organization strategies.

Strategy: The strategy for introducting Ada will vary with the individual service or agency. Although there is a long term goal to adopt Ada as the common language for embedded computer applications, each component has specific software needs peculiar to itself such as a current commitment to other languages with different support systems. The introduction strategy for Ada will depend on the stability and sophistication of those systems. A successful introduction requires a thorough understanding of existing software practices and procedures. As part of the introduction plan, the AJPO and the services must develop the infrastructure for the configuration control,

distribution and maintenance of Ada software, while phasing out existing software development/maintenance practices and procedures. Each organization must ensure the availability of qualified, well trained Ada government and contractor personnel.

In general, introduction of the language should focus on new programs rather than modifications of existing systems. Programs coded in another language should not undergo a translation unless there are compelling economic or performance advantages to this transition. Although mechanical translation technology can accomplish a significant portion of the recoding, the more important advantage -- a structured design using principles of modern programming available through Ada -- would probably not be realized. On the other hand, during a major system or subsystem upgrade, where redesign and translation may offer an effective transition strategy. This is a particularly suitable strategy for Ada since its modular nature facilitates the modification of existing systems by the addition of Ada packages. The best technical solution to the problems associated with mixing languages will be sought to permit Ada procedures to call existing programs written in FORTRAN, JOVIAL, CMS-2, etc., and vice versa.

Progress: On 10 June 1983, Dr. Richard D. DeLauer, Under Secretary of Defense in Research and Engineering, issued a memorandum stating that "the Ada programming language shall become the single, common computer programming language for defense mission-critical applications. Ada shall be the programming language 1 January 1984 for programs entering advanced development and 1 July 1984 for programs entering full-scale engineering development."

The Army plans to use Ada for all Battlefield Automated Systems starting in 1984. The Air Force plans to use Jovial J73 for avionics until an APSE is ready and then to use Ada at low program risk. The Navy plans to use CMS-2 through 1985 and then to use Ada for all new system developments on the AN/UYK-43 and the AN/UYK-44 starting in 1986. The AJPO will coordinate the individual service and organization structure and policies required to provide the necessary infrastructure to support Ada, APSEs and modern software practices.

4.1.2 Develop Hardware Configuration Strategy

Strategy: Advances in hardware technology both for software development and maintenance support and for target software support must be considered. The strategy for small computers must consider the ability of the hardware to support Ada within the constraints of limited memory as long as that limitation exists. The Ada concept of standardization proscribes no subsetting of Ada, but recognizes that the hardware used in small embedded computer applications may not support the entire language. If the AJPO ignores the small computer industry, then defacto subsets of Ada may appear and the AJPO may have difficulty dealing with these pre-established subsets. The AJPO should not relinquish control in this area since DoD applications may be involved. The DoD should take the lead and develop a policy for the development of Ada products for small computers.

Progress: In the small machine category, a number of firms such as Telesoft, Western Digital and others have early, albeit incomplete, implementations on 16 bit machines. It appears that these developments may be successful. Although there is no apparent need for specific action at this time with respect to

16 bit machines, commercial developments cannot be depended upon to solve the software problems associated with the hardware of DoD computer systems. Accordingly, a study of the requirement to support Ada for various types of hardware will be initiated.

4.1.3 Develop Application Area Strategy

Objective: The STEELMAN requirements document reflects the composite needs of all embedded computer systems (i.e., weapon systems, communcation, command and control, avionics and simulators). Although Ada and its support environment satisfies the basic requirements, new packages and additional tools may be needed for particular applications.

Strategy: For example, command and control systems were considered in developing the requirements for Ada, and the language appears to provide the necessary programming facilities to support command and control applications. Command and control, however, depends heavily on data management. Although Ada has ample facilities for defining and operating new data types, it has not yet been demonstrated that efficient data base management systems (DBMS) can be made available to support those applications areas requiring extensive data functionality.

Ada was designed as a systems language for embedded computers; however, it incorporates many features which would be beneficial to the Automatic Data Processing (ADP) community. Recognizing this fact, the Joint Chiefs of Staff (JCS) have indicated support for the Ada Program and requested that the AJPO investigate the potential applicability of Ada to ADP. Futhermore, recent commercial developments are demonstrating that Ada provides facilties which are useful to the ADP community. Since existing high order languages such as COBOL

are adequate, there is no urgency to promote adoption of Ada in the ADP arena.

Progress: In the ADP application area, the AJPO will develop a plan with the JCS to create the necessary demonstrations to support analysis of the use of Ada in its community. Perhaps the most challenging ADP application is a distributed processing, command and control system. The AJPO is working with the World Wide Military Command and Control System (WWMCCS) Information System (WIS) Joint Program Office to develop the appropriate plans for using Ada.

The Defense Advanced Research Projects Agency (DARPA) and NAVALEX are cooperating in the development of such a distributed database system. Computer Corporation of America (CCA) is under contract to DARPA/NAVALEX to integrate into Ada those DBMS capabilities that would be efficient in distributed processing systems. The first task of this contract is the embedded DAPLEX, a high-level predicate syntax that makes database selection expressions easy and natural to write, in Ada. DAPLEX helps the user maintain a higher data integrity by providing language constructs that support the transaction concept, semantic integrity constraints, and null value handling. The expression-level integration of DAPLEX provides the user with a full, associative retrieval capability coupled with highly readable syntax. AdaPLEX, the result of this effort, will be an Ada-compatible language for programming data base applications. AdaPLEX will be passed through a pre-processor to generate Ada code. It is scheduled to be ready for ALPHA testing by the end of 1984. In addition, CCA is to design a distributed implementation within the same time frame.

4.2 DoD Software Policies

Objective: Software policies which mandate realistic objectives for the application of Ada and its support environment are critical to the success of the Ada Program.

Strategy: These policies will establish the infrastructure and management structure within which the software engineering principles of the Ada program will be used. The AJPO, in coordination with the services, will develop or modify existing software introduction, application, design, configuration control, maintenance and distribution policies.

Progress: A draft revision to DoDI 5000.31 added Ada to the list of approved higher order languages and established policies regarding its use with embedded computer systems. For example, the Army will use the ALS for all new systems and major enhancements starting in 1984. Contractors must guarantee that their application software can be maintained on the ALS. Although the DoD will continue to purchase software from many vendors, central facilities using DoD APSEs will maintain the software. Following this concept, the Army has established Post Deployment Software Support (PDSS) Centers for each battlefield functional area. These PDSS centers will use the ALS to maintain the fielded software.

The Navy and Air Force Ada software development and maintenance policies are currently being formulated. Although the mandated date for introduction will be later than 1984, the structure and content of the Navy and Air Force Ada policies will be similar to the Army's.

4.3 Facilitate Use of Ada

Objective: Removing institutional barriers to the adoption of Ada will speed the acceptance of Ada, its environment, and modern software practices.

Strategy: It is likely that potential users who are not familiar with the advantages of Ada may either resist change or design new systems in Ada using old practices. This would hinder the introduction of Ada and negate many cost and quality goals of the program. Providing information to and interacting with the software community will increase everyone's awareness of Ada's benefits and provide program managers with the knowledge required to take full advantage of Ada's technology.

4.3.1 Develop Information Exchange Strategy

Objective: One of the major responsibilities of the AJPO is active dissemination of Ada related information.

Strategy: Increasing the software community's awareness of the benefits of the Ada program will speed its adoption. Timely, accurate, available information will eliminate many misconceptions which are natural by-products of introducing any new technology. A strategy must be devised to identify and provide information to appropriate DoD and contractor personnel.

A feedback mechanism reporting detected Ada implementation errors, proposed improvements, etc. will be provided as part of the information collection process. Feedback from the software community will give users a sense of active participation in the Ada program and provide a channel for constructive criticism of the program and its products.

Since APSEs reflect state-of-the-art software technology, the Ada information dissemination mechanism must prevent the

transfer of support environment technology to potential U.S. enemies. Accordingly, agreements prohibiting the export of this technology to the Warsaw Pact nations or their allies will be made with all parties receiving APSEs. This policy will not pertain to any free-standing, self-hosted Ada compilers.

Progress: The AJPO is establishing dissemination policies and working with appropriate DoD components on Ada export policies.

4.3.2 Information Dissemination

Objective: Active information dissemination to targeted audiences is a major activity of the AJPO.

There are three components in the information dissemination process: data collection, passive information distribution, and active information dissemination to selective high leverage groups within the software community. The Ada Information Clearinghouse will to perform these functions.

An additional responsibility of the information clearinghouse will be to assist the AJPO in preparing Ada program presentations using timely information derived from the Ada information database. These presentations will be given to important components of the software community and government.

A survey will be conducted to determine what information should be contained in the Ada Information Database and the document collection facility. Ada information and documents will be collected and a system installed to automate the process of gathering new information and documents. The information clearinghouse will place copies of all documents in an established distribution facility (e.g., NTIS) and provide periodic updates to that facility. One of the major advantages of placing all the Ada information in an accessible, automated

DBMS is the availability of a system to respond to all requests for information. The Ada Information Clearinghouse will provide convenient public access to all Ada information through their newsletter. Periodic reports will be submitted to the AJPO summarizing Ada activities. The AJPO will use this data as input to the Ada program management decision process.

Progress: The AJPO, as part of its support contract, has established an automated Ada Information Clearinghouse. Ada program information is disseminated through two other mechanisms: (1) AJPO and other government personnel and (2) Ada associates such as SIGAda, (an Ada users/implementors group under the Association for Computing Machinery), Ada Europe, JUG Ada and Ada LETTERS. Ada LETTERS is published bimonthly by SIGAda, and contains articles written by the AJPO and other Ada interested parties describing the status of the Ada program.

4.3.3 Cultivate Affinity Groups

Objective: Acceptance of the Ada program by the software community will be facilitated if Ada receives the endorsement and support of software affinity groups (e.g., academia, users, implementors, management, non-DoD user groups). Active participation by the AJPO in these affinity groups will provide positive interaction and feedback between the AJPO and the software community.

Strategy: Providing information and software to the non-DoD software community will aid in the establishment of Ada as a standard in the non-DoD programming community. It is important to promote the use of Ada and related software standards (i.e., validated Ada compilers, KAPSE standards, Ada style guide,

standard support environment) since this activity will increase the portability and reusability between DoD and non-DoD Ada software.

Process control and commercially embedded software have many of the same software requirements that were the basis for the Ada language design. Furthermore, the quality of Ada software and the life-cycle support provided by the APSE are strong incentives for commercial use. Use of Ada by non-DoD organizations will expand the community, provide infusion of new ideas, and provide a mechanism for the interchange and reuse of software between the DoD and the rest of the software community. Consequently, DoD development costs, in some cases, may be reduced since software developed by non-DoD sources may satisfy DoD's needs. For instance, a private industry produced minicomputer or microcomputer-based system is likely to find wide applicability in service laboratories. A common scientific library would be another area for mutual endeavors.

Progress: Several commercial firms are presently developing Ada compilers. One software vending firm, Intellimac, markets financial systems written in Ada using a commercially developed Ada compiler (TeleSoft). It is anticipated that all major hardware manufacturers will design and develop Ada compilers. Internationally, the German MOD is developing an Ada compiler and minimum support environment. The British MOD has an APSE effort underway and several foreign and domestic industrial firms are evaluating the development of process control and embedded computer systems using Ada. The Swedish Government has announced the development of an APSE residing on the VAX 11/780.

The AJPO will actively encourage such activity, keep abreast of all Ada developments and make relevant information widely available to DoD program managers. The AJPO will

encourage allied countries to enter into agreements to exchange information and complementary software.

4.3.4 Ada In Research

Objective: Introduction of Ada into the research community will be beneficial to the DoD embedded computer community. Steps must be taken to influence and monitor research in various languages. The use of Ada as an implementation language in research projects aimed at the development of advanced software technology should be encouraged.

Strategy: Research use of Ada is advantageous to the DoD for two reasons: first, researchers will become more productive using Ada and its programming support environment and second, since many researchers have contacts with the academic community, they will promote the teaching of Ada-based software engineering principles in academia. It is anticipated that researchers who recognize the advantages of Ada will teach their students the concepts of the language and its support environment. These students will become part of the broad programmer and software engineering base required for Ada to become the DoD standard for embedded computer systems. The DoD-supported research community is a likely source of advanced tools. Every effort should be made to encourage the use of Ada where appropriate. Caution must be exercised so that creativity is not discouraged or the objective interpreted in such a way as to stifle further research on language related issues.

Progress: The AJPO, in cooperation with DARPA, the Office of Naval Research (ONR), the Air Force Office for Scientific Research, the Army Research Office, and other DoD research offices, is promoting the use of Ada in research. The AJPO will

monitor these research activities and coordinate its efforts with the National Science Foundation.

4.4 Incentives

Objective: Facilitating the use of Ada and providing information dissemination are predominately passive devices for motivating the use of Ada. An active mechanism must also be installed to provide incentives for the embedded software community to adopt Ada, its support environment and accompanying software engineering principles.

4.4.1 Incentive Strategies

Objective: The early use of Ada will require a commitment by several service program managers to accept the problems and potential delays associated with its use.

Strategy: Incentives must be developed to overcome the natural reluctance of service managers to accept the additional risk inherent in the application of a new technology such as Ada. The decision to use Ada early will result in lower life cycle system costs; however, it may require greater development expenditures/schedule impacts than those incurred by other languages.

The AJPO, in cooperation with the service Ada program managers, will first identify the types of incentives to be instituted and then target the incentives to those program developers whose early use of Ada would be the most beneficial to the program.

Progress: The AJPO will work with the services to develop an appropriate incentive strategy.

4.4.2 Incentives For Use/Teaching

Objective: The academic community is the training facility for future members of the software community.

Strategy: Incentives must be given to academia to ensure expedient development of a broad Ada software community and adoption of Ada by academic organizations. One possible incentive is monetary support, since universities are usually financially constrained. If the DoD funds several universities to develop an Ada curriculum and teaching aids, academia support will be established; widespread use of Ada in academia will be facilitated by the free distribution of its curriculum and software. Another academic incentive is the establishment of exchange programs between industry, government and Ada-knowledgeable academia.

Progress: Peter Wegner of Brown University was funded to contribute to the development of an Ada curriculum. Strategies for providing incentives to the academic community are being developed.

4.4.3 Incentives For Early Users

Objective: The early application of Ada, prior to its mandated use, will provide a means for the DoD software community to become accustomed to the concepts of the language and its support environment. The DoD will facilitate early training and use of Ada.

Strategy: A positive decision to use Ada is reasonable only if the appropriate support systems can be provided within schedule constraints. When Ada and its support environment are sufficiently mature, the AJPO and the services will ensure that if early users have any additional resource requirements, these resources will be provided.

In the near term, the requirements of individual programs may not fall within the framework of the Ada language and its support environment development timeframe. The AJPO and the service Ada program managers will ensure that managers of major DoD programs are aware of Ada's status in relation to their program schedules, assist them in identifying support requirements for the use of Ada and provide guidance for the preparation of procurement specifications consistent with the state of developments.

Progress: Intermetrics and Carnegie Mellon University cooperated in the development of a prototype Ada compiler on a DEC-20 system. The incomplete compiler was installed on the ARPANET at USC in August 1981 as an interim Ada training tool. The NYU Ada translator also served as an early training tool. Commercially there are several firms with incomplete implementations of Ada which can provide training on a variety of computer hardware.

The services, in coordination with the AJPO, are developing criteria for the selection of early users of Ada. Criteria will be established so that maximum constructive feedback is provided by the widest possible community of evaluators while insuring minimum program risk.

4.4.4 Incentives for Development of Reusable Software

Objective: A major benfit of the Ada program will be derived from the development of reusable application and development software.

Strategy: Incentives for developing Ada packages will be instituted to motivate industry. These packages must conform to transportability standards and be designed to meet general needs rather than just a specific project's needs. Reuse of tested

software will increase the reliability of embedded software and reduce costs since this expense will be amortized over several projects. A potential incentive for the development of reusable software would be the requirement of developers to warranty their software. Since this software would be more reliable, contractors would invest in good reusable software to reduce their warranty costs. Another monetary incentive would be to permit contractors to charge a percentage of the original development cost when their software is reused (i.e., royalties).

Progress: Incentive strategies in this area need to be investigated.

4.4.5 Incentives For Use In Research

Objective: The research community must be motivated to use Ada.

Strategy: Research will lead to a broader Ada Software Community. It will also reduce the cost and increase the quality of research projects. This will be an important factor when research is transitioned to engineering development and eventually production. One possible incentive is to co-fund research projects using Ada.

Progress: DARPA and service research organizations are promoting the use of Ada in research. The AJPO is co-funding a software metrics effort with ONR. This contract will research the frequency of occurrence for various Ada constructs and analyze the programming style of personnel having different software backgrounds (i.e., assembly language programmer, PASCAL programmer, FORTRAN programmer).

Another research area which warrants further study is distributed processing. The AdaPLEX effort is investigating the

application of Ada DBMS systems in distributed processing. As experience with Ada software is acquired, other application areas requiring additional research will be identified. The AJPO will co-sponsor research efforts in relevant application areas.

GLOSSARY

ACVC	Ada Compiler Validation Capability
Ada IC	Ada Information Clearinghouse
ADP	Automatic Data Processing
AIE	Ada Integrated Environment
AJPO	Ada Joint Program Office
ALO	Ada Liaison Organization
ALS	Ada Language System
ANSI	American National Standards Institute
APSE	Ada Programming Support Environment
ARTOS	Ada Run-Time Operating Systems
AVO	Ada Validation Organization
CAIS	Common APSE Interface Set
CEC	Commission of the European Community
CREASE	Catalog of Resources for Education in Ada and Software Engineering
DARPA	Defense Advanced Research Projects Agency
DBMS	Data Base Management Systems
DCA	Defense Communications Agency
DIANA	Descriptive Intermediate Attributed Notation for Ada
DoD	Department of Defense
DoDD	Department of Defense Directive
DoDI	Department of Defense Instruction

DUSDRE(AM)	Deputy Under Secretary of Defense for Research and Engineering (Acquisition Management)
ECS	Embedded Computer System
E&V	Evaluation and Validation
FIPS	Federal Information Processing Standard
FSD	Formal Semantic Definition
HOL	High Order Language
HOLWG	High Order Language Working Group
INRIA	French Government Laboratory for Information Processing
ISA	Instruction Set Architecture
ISO	International Organization for Standardization
JCS	Joint Chiefs of Staff
KAPSE	Kernel Ada Programming Support Environment
KIT	KAPSE Interface Team
KITIA	KAPSE Interface Team - Industry/Academic
LRM	Language Reference Manual
MAPSE	Minimal Ada Programming Support Environment
MCCISWG	Military Command and Control Information System Working Group
MCCR	Mission Critical Computer Resources
MCCS	Mission Critical Computer System
MCT	Methodologies Coordination Team
MIL-STD	Military Standard
MOA	Memorandum of Agreement

MOD	Ministry of Defence
MSC-ECR	Management Steering Committee - Embedded Computer Resources
NBS	National Bureau of Standards
NOSC	Naval Ocean Systems Command
NTIS	National Technical Information Service
ONR	Office of Naval Research
OSD	Office of the Secretary of Defense
PDL	Program Design Language
PDSS	Post Deployment Software Support
R&D	Research and Development
SDS	Software Development Standard
SEEDWG	Software Engineering Education Working Group
STARS	Software Technology for Adaptable Reliable Systems
STI	Software Technology Initiative
V&V	Verification and Validation
WIS	WWMCCS Information System
WWMCCS	World Wide Military Command and Control System

DISTRIBUTION LIST

Dr. Robert Mathis
Director, AJPO
400 Army/Navy Drive
Room 911
Arlington, VA 22202

Major Al Kopp (20 copies)
Air Force Deputy/AJPO
400 Army/Navy Drive
Room 911
Arlington, VA 22202

DoD-IDA Management Office
1801 N. Beauregard Street
Alexandria, VA 22311

Defense Technical Information Center (2 copies)
Cameron Station
Alexandria, VA 22314

Dr. Thomas H. Probert, CSED

Dr. John F. Kramer, CSED

Catherine W. McDonald, CSED (3 copies)

Mr. Seymour Deitchman, HQ

Mr. Robin Pirie, HQ

Dr. Dan Alpert
Director, Center for Advanced Study
University of Illinois
912 W. Illinois Street
Urbanan, IL 61801

Dr. Barry W. Boehm
TRW Defense Systems Group
MS 02-2304
One Space Park
Redondo Beach, CA 90278

Dr. Ruth Davis
The Pymatuning Group, Inc.
2000 L Street, N.W., Suite 702
Washington, DC 22036

Dr. Larry E. Druffel
Rational Machines
1501 Salado Drive
Mountain View, CA 94043

Mr. Neil S. Eastman
Manager, Software Engineering & Technology
IBM Federal Systems Division
6600 Rockledge Drive
Bethesda, MD 20817

Dr. Charles E. Hutchinson
Dean, Thayer School of Engineering
Dartmouth College
Hanover, NH 03755

Mr. Oliver Selfridge
45 Percy Road
Lexington, MA 02173

END

FILMED

2-85

DTIC