NBSIR 76-1094 (R)

Standards for Computer Aided Manufacturing

John M. Evans, Jr., Ph.D., Project Manager
Joseph T. O'Neill
John L. Little
George E. Clark, Ph.D.
James S. Albus, Ph.D.
Anthony J. Barbera, Ph.D.
Bradford M. Smith
Dennis W. Fife, Ph.D.
Elizabeth N. Fong
David E. Gilsinn, Ph.D.
Frances E. Holberton
Brian G. Lucas, Ph.D.
Gordon E. Lyon, Ph.D.
Beatrice A. S. Marron
Mable V. Vickers
Justin C. Walker

Office of Developmental Automation and Control Technology
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D. C. 20234

Fourth Interim Report

January, 1977

Prepared for
Manufacturing Technology Division
Air Force Materials Laboratory
Wright-Patterson Air Force Base, Ohio 45433

NBSIR 76-1094 (R)

STANDARDS FOR COMPUTER AIDED MANUFACTURING

John M. Evans, Jr., Ph.D., Project Manager
Joseph T. O'Neill
John L. Little
George E. Clark, Ph.D.
James S. Albus, Ph.D.
Anthony J. Barbera, Ph.D.
Bradford M. Smith
Dennis W. Fife, Ph.D.
Elizabeth N. Fong
David E. Gilsinn, Ph.D.
Frances E. Holberton
Brian G. Lucas, Ph.D.
Gordon E. Lyon, Ph.D.
Beatrice A. S. Marron
Mable V. Vickers
Justin C. Walker

Office of Developmental Automation and Control Technology
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D. C. 20234

Fourth Interim Report

January, 1977

Prepared for
Manufacturing Technology Division
Air Force Materials Laboratory
Wright-Patterson Air Force Base, Ohio 45433

U.S. DEPARTMENT OF COMMERCE, Elliot L. Richardson, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

TABLE OF CONTENTS

INTRODUCTION

The Air Force is initiating a major new program to accelerate the establishment of Integrated Computer Aided Manufacturing (ICAM) in discrete part batch manufacturing industries in the United States, especially in the aerospace industry. The National Bureau of Standards is providing support to that program by analyzing existing standards relevant to Integrated Computer Aided Manufacturing.

This document is the fourth interim report to the Air Force Manufacturing Technology Division of the Air Force Materials Laboratory at Wright-Patterson Air Force Base on the ICAM support project. This report covers Task 5 of the project:

Task 1 Identify current standards applicable to CAM.
Task 2 Analyze existing formal and de facto standards.
Task 3 Assess the actual usage of standards in industry.
Task 4 Recommend optimal standards for CAM system development.
Task 5 Identify standards organizations and outline a proper Air Force role in standards activities.

The first report covering Task 1 identified those existing and potential standards which will be useful to the Air Force in the development and implementation of integrated computer aided manufacturing systems. Such systems, when implemented by the Air Force and by Air Force contractors, will increase productivity in discrete part batch manufacturing by several thousand percent.

The second report provided a comprehensive reference data base on all formal and de facto standards that are considered to be relevant to the Air Force Program. The report took the form of an annotated bibliography with data sheets on each standards activity for ease of reference. This report covered Tasks 2 and 3.

The third report covering Task 4 discussed the utility of these standards to the Air Force Program and in each relevant standards area recommended a best approach to follow either toward adopting existing standards or toward developing needed standards.

The fourth report outlines the proper role of the Air Force in standards activities. Recommendations are made for a comprehensive and rational approach to computer integrated manufacturing based upon the use of formal standards, definitive technical guidelines, and precise ICAM program policy. These recommendations are made in a framework of programmatic objectives that NBS believes to be essential for the success of the ICAM program.

The work reported here was supported in part by the Air Force Program for Integrated Computer Aided Manufacturing, Manufacturing Technology Division, Air Force Materials Laboratory, Wright-Patterson Air Force Base under MIPR FY 14577600369, Dennis Wisnosky, Program Manager.

THE AIR FORCE ROLE IN STANDARDS

The goal of complete computer integrated manufacturing throughout the aerospace industry requires the interaction of a multitude of functional modules some of which are already in limited use, some just conceived, and others not yet developed. Coordinating the development of this variety of technologies from an equally large number of different contractors and doing it in such a say as to have each module fit into the architecture of an integrated CAM system would appear to some to be an impossible task.

The goal, however, becomes achievable when approached through the creative use of formal standards, definitive technical guidelines and precise ICAM program policy. These three techniques will allow the complete definition of interfaces between ICAM modules and their environment. Where interfaces match, modules can fit together and system integration becomes possible.

Standards are an essential component of the ICAM Program; essential for the development of the ICAM modules, essential for their successful integration, and essential for encouraging the widespread use of the products developed. They are primarily an arbitrary solution to a recurring problem. Therefore, it is important that any standards evolved by ICAM be relevant to the solution of problems as they are normally perceived and acted upon by various classes of users. Equally important is that they be sufficiently well documented that they can be easily understood and implemented by potential suppliers of the products.

It should be remembered that the standardization of a product, CAM module, interface or technique is infinitely more important than standardization on it. In other words, the establishment of an acceptable functional definition of the product is the most difficult and meaningful part of standardization. If this part is done well, that is if it results in broad implementation and use, then informed management decision policy can be formulated. Policy can be set regarding the strictness or looseness of the standards enforcement and the degree to which a product's use must be considered mutually exclusive in relation to the use of similar products. Therefore, standardization effort in the ICAM Program must be viewed as an activity which will create management options by making nonstandard things more understandable, usable, and controllable rather than an activity which limits options.

ICAM Standards Office

So vital are standards to the successful realization of ICAM program objectives that a Standards Office should be created as a focal point for coordination, documentation, and standardization activities. Functions of this office would include:

- Coordination with the NASA effort on Integrated Program for Aerospace Vehicle Design (IPAD) to mutually define the CAD/CAM interface;
- Cooperation and support of various voluntary standardization activities and of the development of FIPS and MILSPEC standards when necessary;
- Development of local ICAM program guidelines where necessary for areas such as documentation and programming standards;
- ICAM data base management system administration;
- Maintenance and distribution of system definition, software tools, and documented ICAM software.

All subsequent recommendations assume an ICAM Standards Office. The minimum staff of such an office would be three professionals: one responsible for the CAD/CAM interface, one would be the Data Base Administrator, and one responsible for maintenance of the system definition and documented ICAM software. Additional clerical staff may be needed as the ICAM program matures.

The ICAM Standards Office will have to contend with three general types of relevant system standards, namely,

True standards that exist and must be maintained, improved, corrected, reimplemented, tested, documented, etc.;

Those that do not exist and that therefore must be created, then maintained;

De facto standards which are really not standards at all but which work and therefore require practical consideration.

All three types will require a commitment of resources if they are to remain or become responsive to ICAM requirements. Table 1 summarizes the various standardization efforts to direct interest to the ICAM program and the degree of Air Force commitment which is recommended by NBS. This commitment need not be of direct involvement of the program staff. Rather, it would be wise and more effective to arrange expert technical representation on the various standardization activities. The technical representatives could intelligently and authoritatively express ICAM needs and be in a prime position to analyze the best use of the developed standard in ICAM products. This function would be given the emphasis of a primary mission and not as a collateral duty assigned to a member of the ICAM staff.

Active Standards Organizations

Voluntary industry standards for products and engineering procedures are set by consensus agreement among concerned parties. There are now some 20,000 voluntary standards in force which were created by more than 400 organizations, covering a multitude of products, practices, test procedures, materials, and other characteristics which were found to be in the interest of parties involved to reach a common understanding and a common practice.

The driving force behind voluntary standards is economic. In a study that was done in the 1920's on standardization during World War I it was found that the cost of production and distribution could typically be reduced as much as 50% through standardization. Cost figures that are comparable to that figure are possible in dealing with computer aided manufacturing systems.

Standards for numerical control and CAM are voluntary standards. The second interim report has identified 64 standards which apply to computers in manufacturing. These standards have been developed by several different organizations:

Electronic Industries Association

The EIA has an engineering committee (IE-31) on numerically controlled machines. This committee has been in existence throughout most of the history of numerical control, and is responsible for those standards that will stimulate the interchange of information between different NC systems, particularly paper tape size, command and data formats, and electrical interfaces between numerical controls and machine tools. The committee has attempted not only to standardize de facto marketplace practices, but to play a role of leadership by issuing bulletims that point out preferred directions for design and implementation of various aspects of machine tool control systems.

Table 1 - Participation on Standards Groups

Technical Area	Standards Group	Degree of Participation	Technical Objectives
COMMUNICATIONS CODES	ANSI X3.4	Monitor	Maintain awareness of basic ASCII code.
PROGRAMMING LANGUAGES			
FORTRAN	ANSI X3J3	Monitor	Insure 1977 revision is made available from ICAM.
COBOL	ANSI X3J4 FIPS TG-9	Monitor Monitor	Extract methodology and guidelines for input to programming policy.
PL/I		Monitor	Define useful subsets to run on smaller computers.
PASCAL			
DATA BASES			
CODASYL	ANSI/X3/SPARE	Monitor	Identify the need for ANSI Standards in DBMS.
	FIPS TG-24	Active	Provide ICAM requirements for DBMS in a distributed system environment.
NC LANGUAGES			
APT	ANSI X3J7	Active	Influence Complete Specification of language for portability.
	FIPS TG-19	Active	Develop Federal Standard and guidelines for use.
COMPACT	ANSI X3J5	Review	Maintain awareness and watch for CLTAPE output.
CAD/CAM INTERFACE			
Physical Object Description	ANSI Y14.26	Monitor	Analyze for use in ICAM's interface with Design.
	CAM-I Geometric Modeling	Monitor	
Electronics	IPC	Review	Evaluate developments for use with Avionics systems.
General	NASA IPAD	Monitor	Resolve any incompatibilities which are noted.
COMMUNICATIONS			
Data Terminal Equip	IE-30	Active	
Peripheral Interfaces	ANSI X3T9	Active	
Link Level Protocols	ANSI TG-4 X3S3	Monitor	
Packet Network Protocols	ANSI X3S3	Monitor	
Host to Host Protocols	None		
DNC Interface	IE-31	Active	Help refine interface descriptions.

The EIA committee has set standards on character codes; formats for numerical control tapes; and interfaces between numerical controls and machine tools, peripheral equipment, and higher level computers.

National Machine Tool Builders' Association

The NMTBA develops various publications relevant to the manufacture, procurement and utilization of machine tools. These include a Directory of NC Machine Tools and Related Products, an NC Character Code Cross Reference Chart, a Definition and Evaluation of Accuracy and Repeatability of NC Machine Tools, Common Words as They Relate to NC Software, and Guidelines for the Measurement of Machine Tool Utilization.

Note: For a listing of standards on the components of machine tools and on the design, manufacture and sales of machine tools, the reader is referred to the Standards Index published by NMTBA.

American National Standards Institute

The American National Standards Institute is the nationally recognized coordinator of voluntary standards development and the clearinghouse for information on national and international standards. Its federated membership includes some 180 voluntary organizations representing virtually every technical discipline, every facet of trade and commerce, organized labor, and consumer interests. These organizational members join with some 1000 individual companies, representing both large and small business, and with representatives of government at federal, state, and local levels in programs dedicated to meeting this country's needs for national consensus standards through voluntary action.

ANSI's approval procedures for recognizing standards as American National Standards ensure a consensus of affected interests. Its requirements for due process and the right to appeal actions at several levels of review establish confidence in, and credibility for, the standards it approves. It provides and administers the voluntary system through which standards, no matter what their origin, may be recognized and accepted nationally and internationally.

Whereas anyone may submit proposed American National Standards to the Institute, the Institute recognizes only three methods for the development of evidence of consensus for approval of American National Standards. These are the Accredited Organization Method, the American National Standards Committee Method administered by the ANSI X3 Committee, and the Canvass Method.

Computer Aided Manufacturing-International

CAM-I is a non profit organization devoted to advancing the state-of-the-art in computer aided manufacturing. CAM-I has an active standards committee. Current projects include a glossary of CAM terms and an analysis of the architecture of CAM.

International Organization for Standards

The ISO is a worldwide federation of national standards institutes (ISO Member Bodies). The work of developing International Standards is carried out through ISO Technical Committees. Every Member Body interested in a subject for which a Technical Committee has been set up has the right to be represented on the Committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. The United States is one such Member Body and is represented on the various ISO committees by ANSI.

Draft International Standards adopted by the Technical Committees are circulated to the Member Bodies for approval before their acceptance as International Standards by the ISO Council. ANSI generally adopts in total those ISO standards it feels have merit. It is recommended that the Air Force should consider ISO standards for informational purposes only and not for adoption.

International Consultive Committee on Telegraph and Telephone

CCITT is the French abbreviation for the International Consultive Committee on Telegraph and Telephone which sets standards primarily in the field of communications. In most nations of the world CCITT recommendations are given the force of law. This is not true of the USA. The CCITT is an organ of the International Telecommunications Union (ITU) which is reported to be the oldest international standardizing body in the world. The ITU is now an organ of the United Nations. The USA is represented on the CCITT by the the US State Department in contrast to the ISO where the USA is represented by ANSI.

Society of Manufacturing Engineers

The SME has a standards committee that is currently evaluating standards for computer aided manufacturing systems. Present projects include a newsletter of CAM standards and an analysis of the architecture of CAM. SME publications include technical papers, reports and text books on NC and CAD/CAM.

Numerical Control Society

The Numerical Control Society Standards Committee has been active in identifying and promoting those standards which apply to NC and CAM. While the committee does not write or set standards, they have recently published a directory of standards for NC machine tools and for the use of computers in manufacturing.

Aerospace Industrial Association

The AIA sets standards relevant to aerospace manufacturing. These standards are issued as National Aerospace Standards (NAS) and include machine tool specifications, systems configuration specifications, and standards adopted from EIA and ANSI.

Department of Defense

DoD writes the bulk of standards that influence Federal procurement of NC and CAM equipment. These military standards and military specifications primarily cover machine tool specifications. DoD issued military standards are available from the Navy Publications and Forms Center.

National Bureau of Standards

NBS administers the Federal Information Processing Standards (FIPS) Program which develops standards for Federal procurement and use of computer systems. Whenever possible these FIPS standards adopt existing ANSI voluntary industry-user standards. However, the FIPS Program may develop in house standards or cite those developed by other industry standards groups to satisfy the requirements of the Federal ADP Community. The FIPS program is organized around interagency Task Groups each of which addresses a single standardization area. Currently there are 16 Task Groups. Membership is open to all agencies of the US government. Draft standards developed in the Task Groups are submitted to NBS for coordination with all Federal agencies and with the public and industry. Final approval is by the Secretary of Commerce on behalf of the President. FIPS standards are mandated in all Federal procurements and can only be waived when justified by the head of an agency. Copies of standards are available from the National Technical Information Service of the Department of Commerce.

Programmatic Objectives

In developing a proper role for the Air Force in this area, it must be recognized that a standard is never an end in itself, but rather a means to an end. Therefore, a careful mapping must be made of program objectives and the manner in which these objectives become more attainable through the creation and adoption of certain standards. The National Bureau of Standards endorses the Air Force view that there are four programmatic objectives that are essential to the success of the ICAM program.

Integratable Modules	Distributed Data Processing
Portability of Software	Exchangeable Manufacturing Data

These programmatic objectives, diagrammed in Figure 1, will strongly influence the Air Force role in ICAM standardization.

Each objective will now be examined in detail with specific recommendations made regarding required standards, guidelines, policy and new developments.

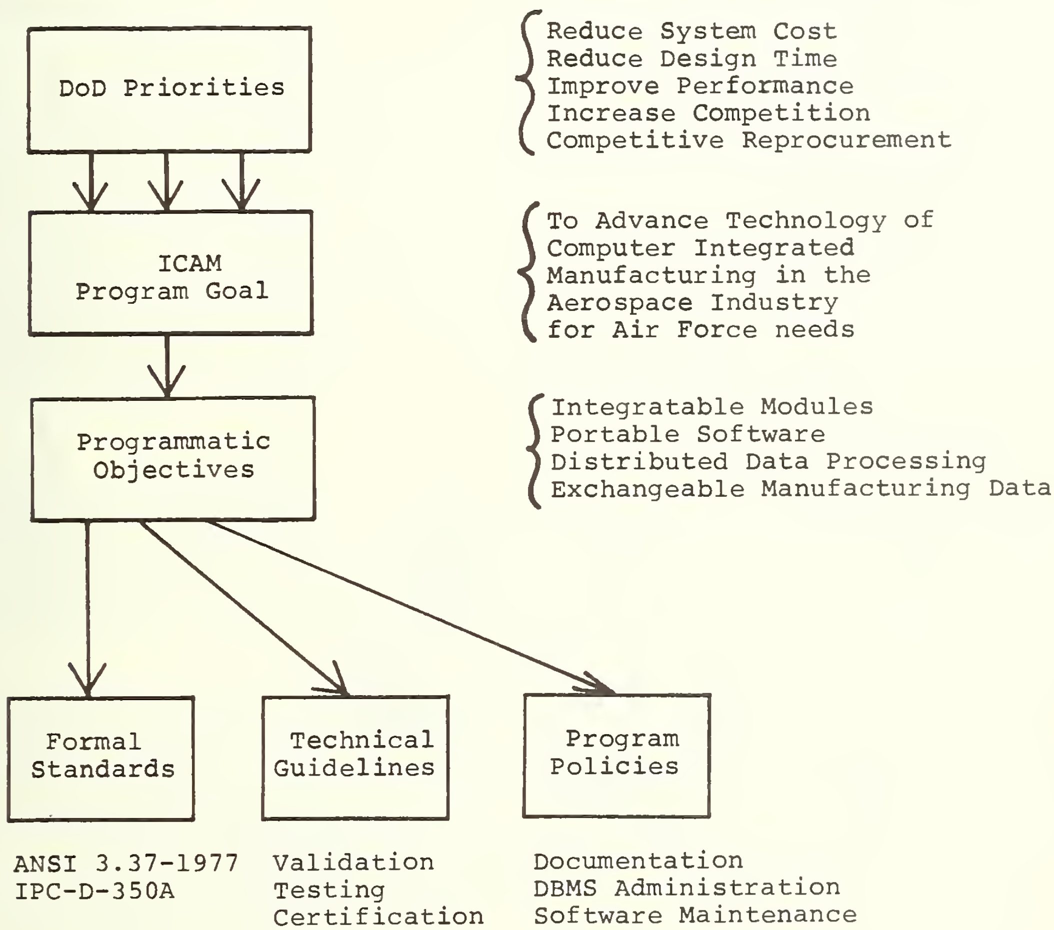


Figure 1 - ICAM Programmatic Objectives

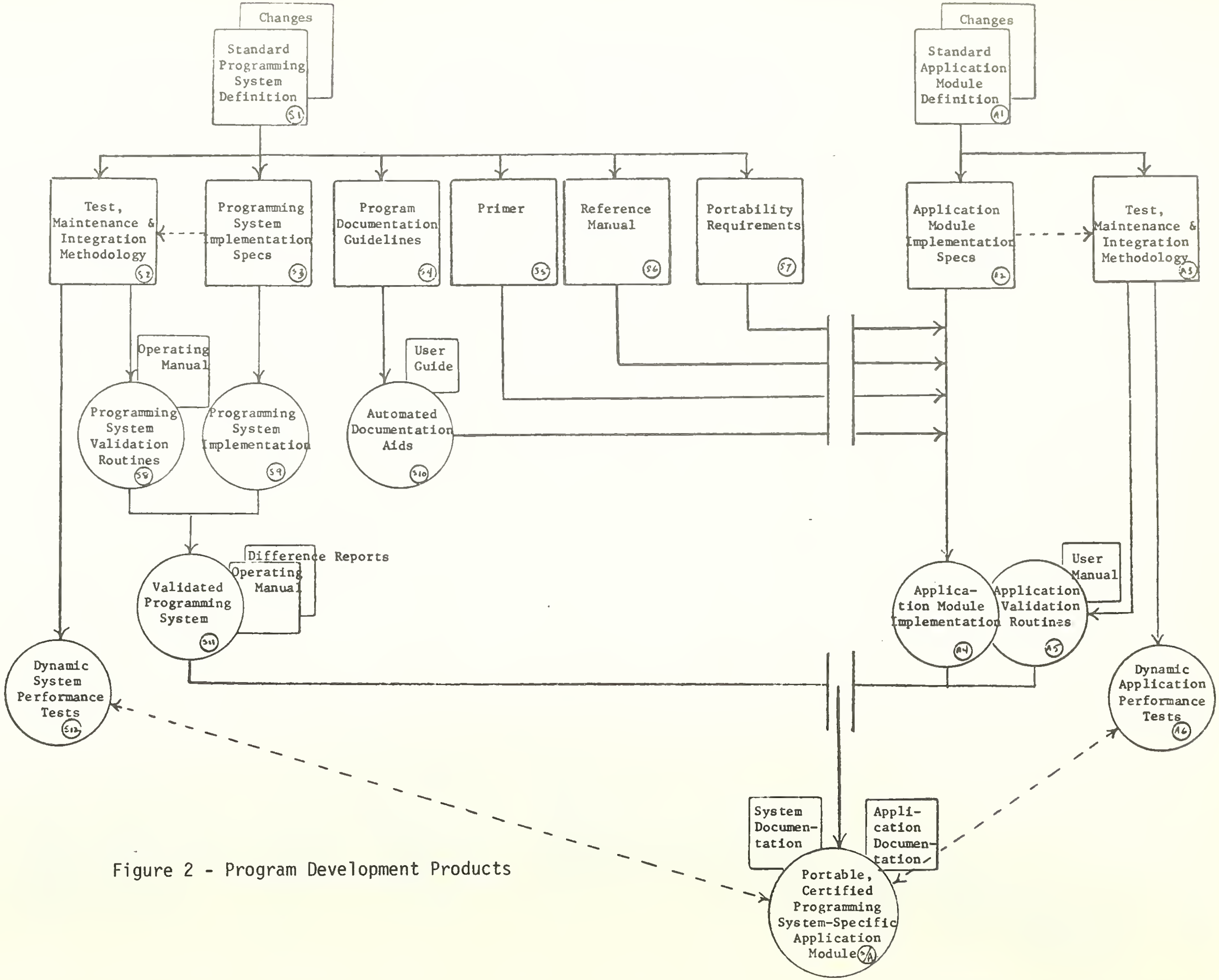


Figure 2 - Program Development Products

PORTABILITY OF SOFTWARE

The ability to transport computer programs among different computer systems with a minimum of software engineering is an essential component of the ICAM Program; essential for achieving the Air Force objectives to stimulate the widespread use of integrated CAM systems and essential for meeting the Department of Defense objectives to provide good technology transfer.

While many will agree that application program portability is desirable, most will disagree on how to achieve it. Obviously the specifying of a few ANSI standards will not produce the desired result. Many more elaborate schemes have been tried -- most having only limited success. Some have sought a common subset of a programming language which will execute on a variety of different configurations. One shortcoming of this approach is that it results in such a limited language set that coding becomes very inefficient. True portability of applications programs without sacrificing language power demands attention to minute detail in the definition, specification, documentation and testing of both system and application software.

Software Development Philosophy

Such an approach was used very successfully with the MUMPS programming system. It evolved over a three year period from several proprietary language dialects into a widely used, public domain programming system capable of easily transportable applications programs among 23 configurations of 15 different computer manufacturers. Experience gained by NBS personnel from the MUMPS success story has been formalized and tailored to address Air Force needs for an ICAM software development philosophy to achieve application program portability. Figure 2 depicts the major segments of this philosophy with the double vertical lines indicating the separation of system functions on the left from the application program functions on the right. The model's notation uses squares to indicate a document or set of papers and circles to indicate a computer program. Vertical lines indicate dependence of a lower entry upon a higher one. Horizontal lines indicate influence or interaction.

Figure 2 is intended to present a model of a software development philosophy which summarizes the processes and products that should exist or be brought into existence for the orderly development, testing, and management of applications programs such that they will be transportable across different computer lines. As this model represents only a single programming system, there will be one such model for each language used, i.e., FORTRAN, COBOL, PL/I, etc.

The system software which results from the definition, specification, implementation and testing on the left can be viewed as an "aid" to the development, testing, and operation of the applications modules shown on the right. Since applications programs must be expressed in a language that is acceptable to the computer, the construction of the system software which interprets that language becomes of critical importance. It defines the media of communication between man and machine, and much of what will be easy or difficult for the man in this relationship will be a direct result of the quality of this system software, its documentation and its techniques for testing and debugging of programs. Each of the functional elements of this software development philosophy will now be examined.

Standard Programming System Definition (S1)

This is the principal system element in the model. It is a meticulous and comprehensive functional definition of the programming language. The term "standard programming system definition" should evoke the image of a public domain document, one that is the result of a consensus arrived at by a representative group of interested users and suppliers, and one that has been used as the basis for successful implementations, hopefully on a variety of computers. Of the hundreds of programming system definitions in everyday use, only a few of them can qualify as standards. To date, only five programming language system definitions have been submitted for consideration as National or Federal Standards: FORTRAN, COBOL, PL/I, MUMPS, and BASIC.

Many of the newer system tools are highly proprietary, and no standards action is currently being taken on them. What this means is that the Air Force must either sponsor a variety of standardization activities that it feels are needed or resign itself to the use of proprietary products and the attendant miseries of sole source procurements, hardware dependencies, and the like. Clearly the route is easier if standardized products are chosen for Air Force use. However, ICAM may not wish to prohibit the use of non standard languages where their capabilities are superior. In these cases an effort should be initiated to formalize the Programming System Definition through a concensus opinion of users and suppliers so that compilers may be implemented on other computers to effect portability.

By way of clarification, the suggestion here is not that end products like compilers, data base management systems, and operating systems, should be in the public domain, but rather that standard definitions, upon which such products can be based, should be in the public domain. This approach is sound from a business standpoint, in that the definitional activity becomes a resource-sharing operation, thereby attracting to the task of orderly system definition, growth, and maintenance many of the most talented and interested individuals from the user and supplier communities. Also the implementation activity is strictly a free enterprise operation in which suppliers are rewarded for their ingenuity and innovativeness, and thereby are encouraged to separately build, from these standard definitions, ever more capable and efficient standard products.

System Validation

Functions in this area represent elements of the first phase of system testing, that of system validation, wherein an assessment is made of the degree to which a programming system implementation such as a compiler (S9) conforms to the standard definition (S1) which it purports to follow. Shown in Figure 3 the actual components of such a test are a supplier-developed programming system implementation (S9) and system validation routines (S8). The routines consist primarily of data and are specific to the standard programming system in question (S1) but are implemented in a general fashion that enables them to interact with any product (S9) that is based on the standard definition (i.e., any product that follows the development plan depicted by the vertical path S1, S3, S9). The validation routines are then "run" together with the programming system implementation, (S9), to provide a pass/fail analysis. Upon completion of the test, the programming system implementation (S9) is classified as either valid (S11) or invalid (in which case it remains an "S9" product). As can be seen in the figure, establishment of programming system validity is crucial to the development of portable application programs, as it provides the single system "link" with application programs (A4), thereby making such development possible.

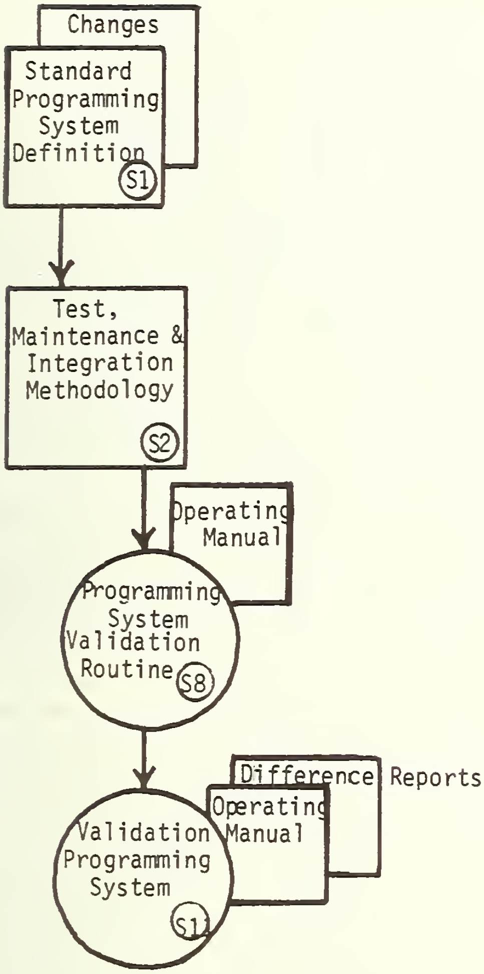


Figure 3 - System Validation

A few words might be said here about the "difference reports" that are shown accompanying the validated programming system (S11). In order to be a constructive aid to standardization, one that will be used by both implementors and users, the validation process should support rather than interfere with the other important processes of system definition, specification, implementation, documentation, and maintenance. A recommended way of doing this is to allow, and in some cases, even encourage differences in all of these areas so long as application program portability objectives are not compromised. The following is an example of how this can be done. Assume that a supplier "extends" his specification (S3) beyond the standard (S1), implements new, nonstandard features, and incorporates them in his proprietary product (S9, S11). The supplier will then want to document these extensions in his instructional literature. However, this is not sufficient for the user who wants to preserve program portability, as the use of these extensions will render his application programs nonportable. Such a user will require notice of the presence of the extensions so that he can invoke a management prerogative to either permit or prohibit their use. Since the notice must precede the choice, the notice should always appear; its ideal place is in a document similar to the portability requirements (S7), whose purpose is to give an "early warning" of impending problems. If an extended system implementation (S9, S11) is chosen for operational use, then the user should write an installation-specific version of the portability requirements document which clearly delineates policy regarding the use of nonstandard constructs.

System Portability Requirements

This document originates in the standard programming system definition (S1) and may even be an appendix to that document. Its purpose is to highlight, for the benefit of implementors and application programmers, aspects of the system that must be accorded special attention if program transferability (i.e., portability of application source code between various system implementations) is to be achieved. It makes an a priori identification of potential problem areas that one could be expected to encounter during the specification phases of standard systems (S3) and standard applications (A2), and issues appropriate cautions regarding the definition and implementation of system extensions and, to application programmers, corresponding cautions regarding the use of such extensions.

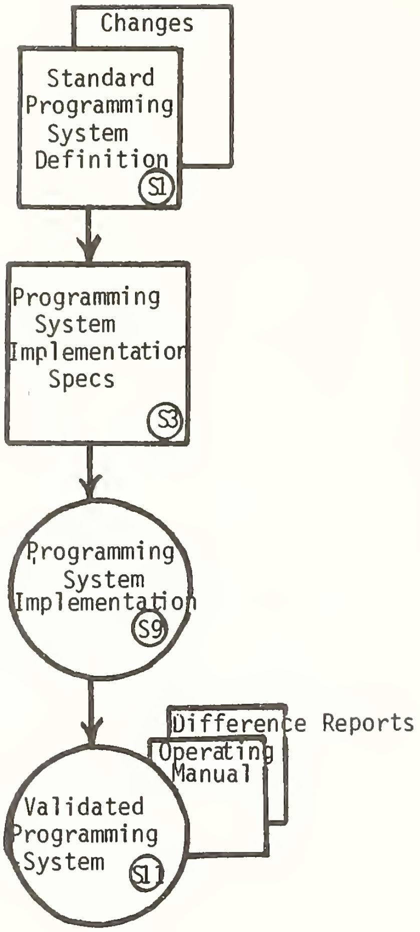


Figure 4 - System Implementation

Programming System Implementation

Products required for the implementation of a validated programming system are shown in Figure 4. Programming system implementation specifications, (S3), are owned, developed, and maintained by the various system suppliers and consist of the standard programming system definition (S1), with or without extensions, and are augmented by configuration-specific implementation instructions. These specifications serve as the basis for an "interim" product, a programming system implementation (S9) and the validation version of that implementation (S11).

System Primer and Reference Manual

A primer (S5) and a reference manual (S6), represent two different levels of training aids, the former for the novice technicial, the latter for the more experienced one. They serve as a bridge between system implementation (S3) and application implementation (A2). Both can be developed centrally as part of the definition phase (S1), thereby enabling potential implementors to share costs, without risk of violating antitrust laws. Furthermore, implementors who find it necessary or desirable to extend their system implementations beyond that of the standard may extend these documents accordingly.

Documentation Guidelines and Aids

Program documentation guidelines (S4) and automated documentation aids (S10), represent two system products that are indispensable to the program preparation process. The guidelines (S4) are specific to the programming system in question, taking into consideration certain eccentricities of that system and the resulting need for lucid representation of logic, data, etc. The automated aids are computer programs that render the system "self-documenting." These are an optional extension to the system guidelines. They encourage and assist programs in the production of detailed, uniform application program documentation.

SYSTEM AND APPLICATION PROGRAM TESTING

Figure 5 shows how criteria used for testing an applications program closes the loop in this philosophy of software development. The testing of the applications modules produced insures that they do in fact conform to the criteria established in the test methodology (S2 and A3). Application Validation tests (A5) certify that an implemented module satisfies the problem to be solved, that it adequately detects and handles all erronious input data and that it is coded in acceptable programming language for protability requirements. Dynamic tests (S12 and A6) are those which probe the behavior of system software or application programs while they are in an operating state. The tests differ substantially from the static tests performed under system or application validation (S8 and A5). Dynamic measurable behavior generally includes timing and memory space utilization.

In brief, dynamic system tests are ones that probe the behavior of system components (e.g., compilers, operating systems, data base management systems, hardware devices, etc.) either individually or in combination; they are "driven" by hooking them on to "live" applications (S/A, resulting from the interaction of S11 and A4) or by hooking them on to simulated applications (S/A, resulting from the interaction of S11 and A6, where A6 is a "rigged" substitute for a live application (A4)). Dynamic application tests, on the other hand, probe the behavior of application programs by isolating and analyzing certain aspects of the "A portion" of the S/A entry.

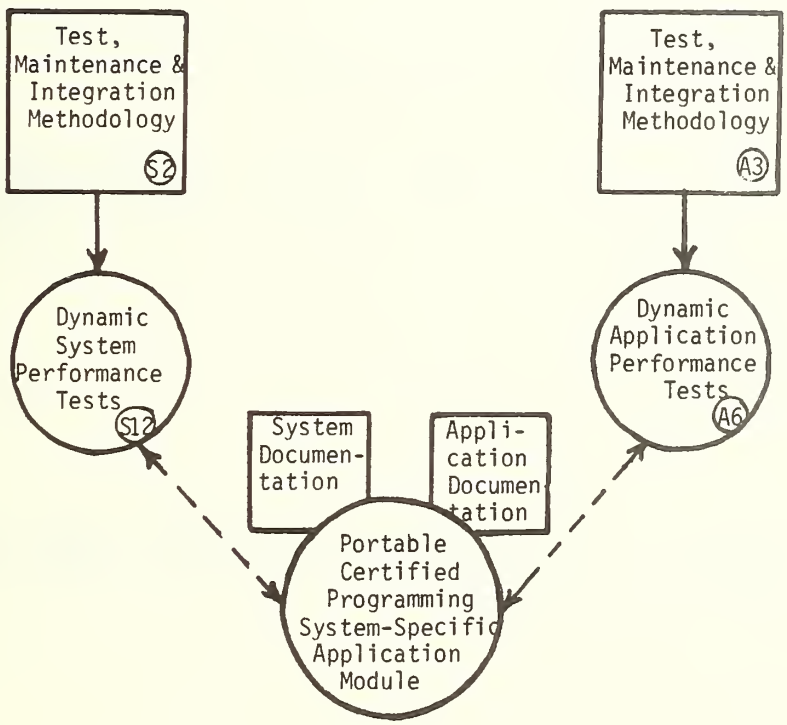


Figure 5 - Program Testing

These dynamic tests provide information that is indispensable to the processes of understanding, maintaining, and improving the performance of both system and application components of complex automated systems.

Summary

Portability of applications programs among different computer systems or configurations is an essential component of the ICAM Program. This portability cannot be achieved without giving detailed attention to both the programming system and the applications program development process. A model of a software development methodology has been presented which addresses the various elements of standards, guidelines, and discipline that are needed by ICAM to permit the development of:

1. portable application program modules,

2. maintenance and improvement methodologies for both system and application programs, and

3. "fall-back" procedures for the eventual replacement of individual system or application modules.

For this methodology to work all of the elements starting with the standard programming system definition must be available to an ICAM contractor. Table 2 shows for the five existing standard system definitions the availability of the system elements. Where none exist, resources will be required for the needed development.

Table 2 - Available Software Development Definitions and Supporting Products

	FORTRAN	COBOL	MUMPS	PL/I	BASIC
Programming System Definition	X	X	X	X	X
System Implementation Specs	X	X	X	X	X
System Validation Routines	X	X	X		X
Applications Validation Routines					
Dynamic Performance Tests	X	X	X		
Difference Reports		X			
Primer and Reference Manual	X	X	X	X	X
Portability Requirements		X	X		X

Air Force insistence on the availability of standard system software and willingness to sponsor the type of effort needed will vastly facilitate solutions to the long term problem of producing portable software for computer integrated manufacturing systems.

RECOMMENDATIONS

(Policy)

1. ICAM should formulate a definitive written policy on how it intends to achieve portability of applications software among different manufacturers and configurations of computer systems. The document should conform closely to the philosophy presented above.

2. Applications programs should be developed only with high level programming languages except in those rare instances where acceptable performance can only be achieved through assembly language. These cases must be carefully controlled and documented.

3. The Air Force should encourage the use of standardized programming languages. NBS believes their effective use to be the key to software portability.

4. While FORTRAN and COBOL will suffice for near term applications, conversion to the use of a more modern language should be encouraged whenever the various standardized supporting products are available to meet ICAM requirements for portability.

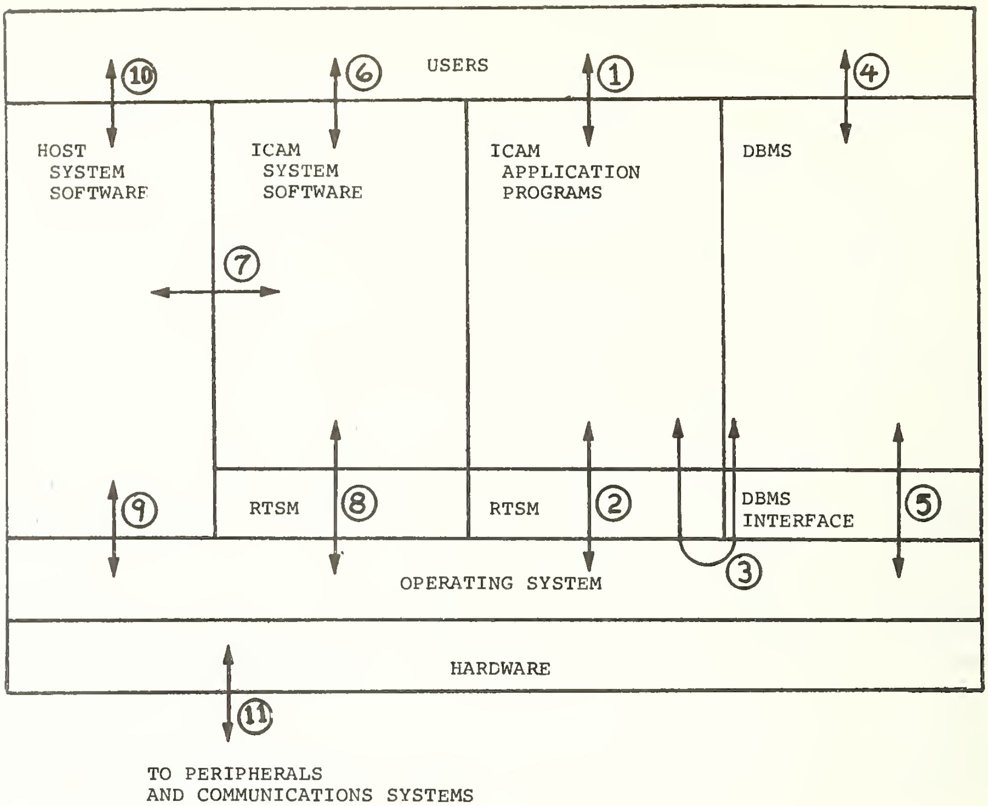
(Guidelines)

1. Guidelines for recommended programming practices are needed to assure that applications code generated will be legible and maintainable.

2. Program documentation guidelines are required to be consistent with the software development philosophy and to assist in the transfer of software technology among different Air Force contractors.

(Standards)

1. The Air Force should support the expeditious establishment of a Federal FORTRAN standard based upon a revised national standard. If ANSI does not approve the revised FORTRAN in 1977, CAM officials should support NBS effort to establish a Federal standard from the best available ANSI proposal.



MAJOR INTERFACES

APPLICATION PROGRAM INTERFACES

1. User Language
2. Run Time Support Monitor: one for each programming language (interprets system calls)
3. Data Manipulation Languages (DML) and sub-schema Data Definition Language (DDL)

DBMS INTERFACES

4. Self contained language queries of data base
5. DBMS/Host interface: designed for each host system ICAM system interfaces (supplied by vendor)
6. User/System commands (machine independent)
7. ICAM System/Host System interface: designed for each host system
8. ICAM System/Host interface: run time support monitor for higher level language used to implement ICAM system

HOST SYSTEM INTERFACES

9. Host System/Host interface implemented by vendor
10. Operating System Command Language (to be avoided if possible)

HARDWARE AND COMMUNICATIONS INTERFACES

11. Interfaces to Networks and Local peripherals

MAJOR ICAM FUNCTIONS

ICAM APPLICATIONS PROGRAMS

Tool Selection
Process Planning
Inventory Control
Etc.

DBMS

Management of all data (distributed)
Query response/report generation

ICAM SYSTEM SOFTWARE

Text Editing
Debugging and Test Software
Math Libraries
Etc.

HOST SYSTEM SOFTWARE

Assembler
Compilers
Linkers
File Management Commands
Etc.

FIGURE 6 - SYSTEMS INTERFACES

INTEGRATABLE MODULES

The development of a true general Integrated Computer Aided Manufacturing System using software developed by several different contractors requires a partitioning of the system into logically separable modules with well defined interfaces.

Figure 6 schematically shows the location of the major interfaces of a large integrated system such as ICAM. A few of these interfaces will be explicitly considered:

- Interfaces between ICAM applications programs and the host system
- Interfaces between ICAM applications programs through the DBMS
- System software interfaces

Applications Program Interfaces

The first of these interfaces is the key to software portability: with adequately standardized and validated programming languages, the applications programs will be essentially independent of the host system. System calls are intepreted by the run time support monitor (RTSM) which is supplied by the host system vendor for each supported programming language. If the language is standardized, the RTSM is also thus allowing portability. The user interface (1) is a real time interactive user language, if any, allowing the user to interact with the system in real time. The location shown for interface (1) is the logical interface; the physical interface is, of course, through a terminal with its hardware interface (11) and the operating system. Note that there are no interfaces between applications programs and the system software.

The language standards relevant to interface (2) are discussed in Portability of Software.

Data Base Interfaces

From the point of view of integration of application modules, the most critical element is the data base. A single (but, perhaps, distributed) data base, to be accessed by all programs through a well defined interface, is the key to the integration of a manufacturing operation into a complete computer-aided system. This data base becomes the source of all information for, and hence the interface between, all applications programs. This critical module - the data base - must be carefully structured and managed. This is the job of the data base management system (DBMS).

Over 200 different DBMS packages are presently available although there exist significant variations in the capabilities and features provided by these systems, certain common functions and interfaces are noted. Applications programs communicate with the data base through interface (3) often with a form of data manipulation language. Interface (5), of course, is essential for interacting with the operation system to store and retrieve the data on the various physical storage devices. This interface sometimes uses a device media control language which is specific to each host computer. Often query packages are provided to users to allow for non-procedural type requests on the data in a self contained query language through interface (4).

No standards exist as yet in the area of data base management systems. The CODASYL specifications have gone the farthest to define and structure an approach which has modular architecture and well defined interfaces. But while there are several CODASYL-type systems available, they are by no means identical and the specifications themselves are still in a state of change. Consequently, there can be no one DBMS identified as being "best" for ICAM use. As a single DBMS will be critical to achieving integration of ICAM software modules, a competitive procurement of a commercial data base software package is recommended to support all near term projects. Emphasis should be placed on obtaining modular architecture, well defined interfaces, portability of applications programs, integratability of ICAM modules, and future adaptability to a computer network system with distributed data bases. The evaluation for selection should include a benchmark demonstration of performance on a typical CAM application.

The key to successful operation of a large DBMS is the Data Base Administrator who is recommended as a key person in the Air Force ICAM Standards Office. The Data Base Administrator:

a. creates and maintains the data definitions for existing application system, and establishes the data definitions for new systems;

b. maintains a library of the data available on the data base;

c. provides data base documentation to the analysts and programmers, such as cross reference listings between data and programs;

d. controls the schema and subschema, thereby controlling access to the data base;

e. is responsible fcr improving the efficiency of data base operations, including performance monitoring activities, keeping statistics on the use of different data, and monitoring the use of physical file spaces; and

f. in general, maintains integrity and security of the data, including definition of backup requirements and recovery procedures.

System Interfaces

Certain system software will, by necessity, be specific to the host computer system and hence will be supplied by the host vendor. In particular, compilers, linkers, accounting programs, and the system executive will all be host specific.

The interface between a user and the host system software is shown as interface (10). Communication across this interface must be expressed in the language of the host system software and hence will vary from system to system. Extensive use of host specific software will endanger system portability and hence use of this interface should be discouraged. A better way of talking with the host system software while preserving portability is through interfaces (6) and (7) and the use of specially developed ICAM system software.

ICAM systems software should contain all of the software tools needed for the development, maintenance and operation of the applications modules on the host computer with the exception of the usual vendor supplied system software mentioned above. This ICAM systems software, written in a high level language so as to relatively transportable itself, will contain such programs as:

Text Editor - For entering, correcting and modifying applications and general text such as program specifications and design documentation.

Program Librarian - For storing all program texts, associated job control statements, common data definitions, test data, and for maintaining a chronological record of modifications between distinct versions. Includes appropriate access controls for members of the project group. This may require a DML interface to the DBMS.

Test and Debug Programs - For analyzing program behavior during preparation and execution on test data input, and deriving execution statistics and traces to help correlate program output with the results of individual high-level language statements. These programs will be specific to a particular high-level language such as FORTRAN.

User Interface - Programs which present a standard interface to the user and translate system commands into host specific commands at interface (7). This is essentially a transparent "shell" that provides a machine independent system executive.

Documentation Aids - Programs to assist in documenting applications programs as they are developed.

Project Manager - For recording chronologically the activity of the individual project members on defined application program modules and deliverables of the project.

Each item of ICAM systems software being written in the high level language interacts with the host operating system through interface (8) in exactly the same manner as an application program does through interface (2).

Peripheral Interfaces

Near the bottom of Figure 6 the interaction of the operating system with the various hardware peripherals is shown by interface (11). Local peripheral equipment interface standards should follow the ANSI standards on the channel level and on the minicomputer device level when they are adopted. Communications interface standards are discussed below under distributed data processing.

Recommendations on Systems Integration

(Policy)

1. The DBMS is the key to integration of applications modules. Functional specifications should be prepared for the competitive procurement of a commercially available data base software package to support all near-term ICAM projects. The specification should require the package to be available on all hardware systems that would be considered for CAM applications in the first few years of the program.

2. A Data Base Administrator should be appointed to the staff of the Air Force ICAM Standards Office. This person should remain fully informed of the state-of-the-art in data base software, and evaluate available packages for future ICAM application and standardization. This person should participate in on-going Federal and national standards efforts, including the NBS FIPS Task Group 24.

3. The Air Force should provide staff to support the CODASYL Task Group in future developments.

4. Dependence on host system software should be minimized, with interfaces provided through machine independent ICAM system software written in a high level language.

(Guidelines)

1. The following user guidelines will be required for effective use of the DBMS. If they are not supplied by the vendor, they should be created by the ICAM Data Base Administrator.

   Data Dictionary
   Device Media Control Language Guide
   Data Manipulation Language Guide
   Guide to the user of query package/report generator

2. A Guide to ICAM systems software should be developed and distributed to all ICAM contractors and users.

(Standards)

1. There are no standards on DBMS or operation systems.

2. Local peripheral equipment interface standards should follow the ANSI standards on the channel level interface and the minicomputer device level interface when they are adopted.

DISTRIBUTED DATA PROCESSING

A stated objective of the Air Force ICAM program is compatability of ICAM software with Fourth Generation distributed system concepts. The trend toward linking central processing units together with either local or commercial communications networks is clear, and the implications for improved efficiency and lower costs is also clear.

The scenario for CAM applications for the 1980's is, then, a series of mini and microcomputers, each running dedicated applications programs, networked together in a total CAM system. One computer may be running the warehouse and providing inventory control, another running CAD programs and supporting interactive graphics, another supporting process planning programs, while an entire hierarchy of computers operates the machine tools and materials handling systems in actual manufacturing tasks. Each computer will keep its own local data base, or possibly have a local "back end" processor running a DBMS, and will be able to access relevant data throughout the distributed system through the communications system by using appropriate access protocols.

The potential power of such an array of processors all working in parallel and yet all acting coherently and with distributed data access is tremendous. Even more importantly, the cost reductions of such a system may be remarkable as mini and microcomputers and mass storage devices continue to drop in cost. Furthermore, the architecture of this systems design ideally satisfies the ICAM objective for having individual modules developed and implemented separately to perform useful work in a stand alone mode while allowing for eventual systems integration.

Subcontractors may be able to make use of CAM capability by directly receiving manufacturing data packages through a network in a form that will directly run their machine tools and inspection machines. Procurement costs will drop significantly if procurement practices are changed to reflect the opportunities of this new technology.

Standards for Distributed Systems

The layers of protocol for one application program in one computer to talk to another application program in another computer are diagrammed in Figure 7. The lower three levels are the subject of formal standards, and the combination of EAI RS232, RS XYZ, CCITT X.21, ANSI ADCCP, and CCITT X.25 offer a comprehensive definition of protocols to access commercial packet communications and switched networks. These standards provide a sound basis for a distributed processing system using commercial communications systems.

The higher level protocols are as yet unstandardized. Host to host protocols are defined by each vendor, for example, in IBM's SNA or DEC's DECNET, but these are specific to each vendor. Potentially a project standard could be implemented for each allowed host system. Process level protocols should be the subject of project standards.

The National Bureau of Standards is initiating (1977) a new project in support of the Air Force Rome Air Development Center to define these higher level protocols. This work will provide a basis for developing ICAM project standards.

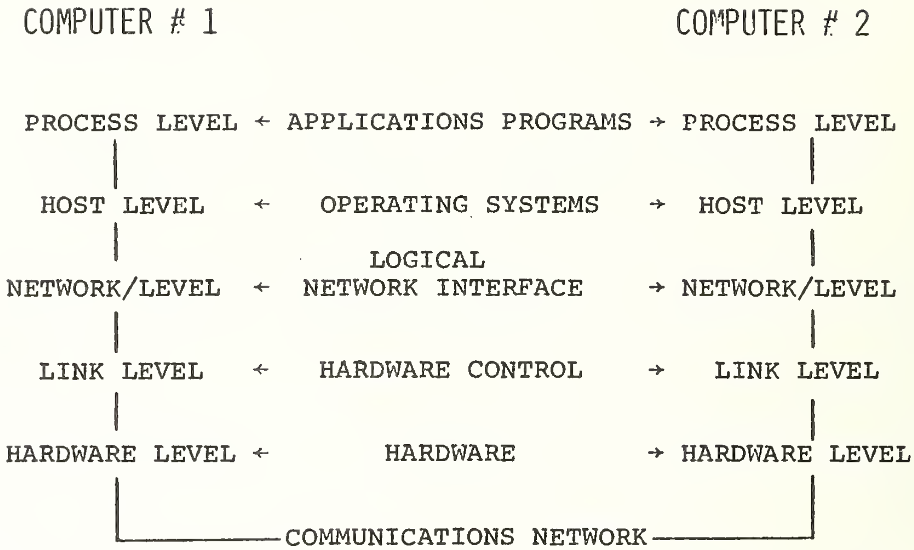

COMPUTER # 1
COMPUTER # 2
PROCESS LEVEL
APPLICATIONS PROGRAMS
PROCESS LEVEL
HOST LEVEL
OPERATING SYSTEMS
HOST LEVEL
NETWORK/LEVEL
LOGICAL NETWORK INTERFACE
NETWORK/LEVEL
LINK LEVEL
HARDWARE CONTROL
LINK LEVEL
HARDWARE LEVEL
HARDWARE
HARDWARE LEVEL
COMMUNICATIONS NETWORK

Figure 7

RECOMMENDATIONS

(Policy)

1. The ICAM system should be designed and implemented in such a way that it could be run on a highly distributed fourth generation computer system in the 1980's. The development and use of communications software and protocols, programming language subsets for minicomputers and microcomputers, and a network model DBMS follow from this approach.

2. The ICAM Standards Office should task NBS to specify host level and process level protocols for use as ICAM project standards, working from the Rome Air Development Center project as a base.

3. The ICAM Program Manager, with advice from the Industry Review Panel should set security requirements for distributed systems.

4. The ICAM Industry Review Panel should be tasked with considering the impact of this technology on procurement practices in contracts between the Government and a prime contractor and between a prime and various subcontractors.

(Guidelines)

1. A guide to the use of DTE/DCE (Data Terminal Equipment/Data Communication Equipment) interface standards, link level protocol standards and network level protocol standards should be developed and distributed to all contractors and users of ICAM software.

2. A guide to security in a distributed system should be written and distributed to all contractors and systems users.

3. A guide to implementing host level and process level protocols should be developed by NBS and distributed to all contractors and systems users.

(Standards)

1. The following standards are recommended for ICAM use:

   DTE/DCE Interface: RS 232C, RS XYZ, CCITT Recommendation X.21
   Link level protocol: ANSI ADCCP
   Network level protocol: CCITT Recommendation X.25
   Communication Codes: ANSI X3.4 (ASCII)

2. The Air Force should task NBS with maintaining cognizance of developments in the recommended standards, representing Air Force interests in standard committee meetings and providing regular reports to the Air Force for distribution to ICAM contractors and users.

EXCHANGEABLE MANUFACTURING DATA

Too often we view the output of the manufacturing process as simply the delivery of finished goods. However, in government procurement of weapons systems the output of the manufacturing process is seen to also encompass the data which defines and controls the manufacturing of the end products. The difference is significant and reflects the concern of the Department of Defense (DoD) with the life cycle cost of the system. Data must be available which completely defines the manufacturing process of each component part of the end product. More importantly, this data must be presented in such a form as to be meaningful and useful to the different manufacturing systems that may be involved in the production of parts for the system over its entire life span. The importance of this exchange of data is illustrated by the following evolutionary phases of a typical system:

Developmental Phase	Part manufacturing data flows to and from Primes and Subs
System Delivery Phase	Manufacturing data on each piece part is supplied to government
Competitive Remanufacture Phase	Government furnishes complete manufacturing data to bidders. Primes may pass data to subs.
Maintenance and Repair Phase	Manufacturing data flows among various government repair facilities and logistics

Engineering Drawings

Presently the bulk of this manufacturing data is exchanged through the use of engineering drawings. Much time and effort at specifying and standardizing this form of data presentation have made it a useful and universal method. Although time consuming and expensive to create, engineering drawings fulfill their purpose well in being easily exchangeable and readily understandable.

However useful engineering drawings may be,they have been a creation of a manual system. Drawings are developed by hand and are meant to be interpreted by hand. Even though computers are assisting in the preparation of some drafting work, the method of data presentation is still pointed toward the exchange of manufacturing data in a conventional environment.

Digital Data Package

The increasing use of computer technology in manufacturing is creating some interesting problems in the exchange of discrete part data. For one thing part data is taking on more forms than just the traditional dimensioned engineering drawing. Digital data descriptions are now being used by numerically controlled machine tools to manufacture parts directly. These digital part descriptions are exchanged in various codes and formats on paper tape, punched cards or disk files. The most prominent standard in this area which pertains to manufacturing data is for the APT programming language; however, a competing language, COMPACT II, is in the early stages of standardization. COMPACT appears to be more efficient than APT for simple parts but is not as capable as APT when used for very complex parts. It should be mentioned that neither language is sufficient as a manufacturing data package. Nor is the APT language in a sufficiently standardized form to serve Air Force needs. While the new standard on APT to be released in 1977 will do much to alleviate this problem, guidelines are needed on the interpretation and use of the APT language to make it a more flexible and versatile tool for exchanging NC manufacturing data among functionally equivalent machines.

Promising work is being done for the manufacturing interface in the CAM-I Geometric Modeling Project, by the ANSI subcommittee on Computer Aided Preparation of Product Definition Data, and by NASA's IPAD Program. The Air Force is advised to continue to maintain close liasion with these three projects as a useful and meaningful manufacturing data specification is surely to evolve from the healthful interaction of these efforts.

Whether one talks of integrated CAM systems or simple numerical control machines, much less reliance is placed upon formal dimensioned drawings. Where drawings are used at all, their format is often simplified to convey general ideas rather than specific detail. Exact dimensions are transmitted digitally.

The trend is clear. As the Air Force and industry progress toward complete computer integrated manufacturing, the dimensioned engineering drawing will cease to be an essential component of manufacturing data. It will be supplanted by a digital part data description that will completely describe the manufacturing process to be used. It will also define digitally any computer generated drawings that are needed for informational purposes. The evolution of this digital part data package is inevitable. While it will eventually affect all components of the Department of Defense, its impact will be felt first in the ICAM Program. Techniques must be developed to allow digital data to be exchanged in as versatile, efficient and flexible a manner as engineering drawings are done now. Technical barriers must be overcome to enable this data to be exchanged among various combinations of government agencies, prime contractors and subcontractors. Means must also be found for the data packages to be exchanged between computer based manufacturing systems and the conventional types. Unless these capabilities are developed, the Department of Defense will incur many unnecessary costs in the evolution of the computer integrated manufacturing systems envisioned by the ICAM Program. The groundwork must be developed now for firm policy, effective guidelines and explicit data standards that will collectively enable the needed exchange of digital manufacturing data.

RECOMMENDATIONS

(Policy)

1. Task the ICAM Industry Advisory Group to develop recommendations for policy regarding the specification, purchase, data rights, and exchange of digital data for discrete part manufacturing.

2. Through the ICAM Standards Office maintain close liasion with the NASA IPAD Program, the ANSI Y14.26.1 subcommittee on Computer Aided Preparation of Product Definition Data, and the CAM-I Geometric Modeling Project to avoid potential problems in compatability.

(Guidelines)

1. Support the development of guidelines on the interpretation, processing and use of the APT language for numerical control manufacturing, such that an APT part program can be quickly and easily exchanged among different machine tools, different shops and different contractors. Inconsistencies in the present use of the language inhibit this ability.

(Standards)

1. Where the Air Force specifies a single part programming system for numerical controlled machine tools, it should conform to ANSI 3.37-1977 APT. If two part programming languages can be acceptable COMPACT II is recommended for efficiency at programming less complex parts provided the CLDATA file can be produced to be compatible with APT output.

2. Through the ICAM Standards Office task the National Bureau of Standards to maintain close liasion with the ANSI subcommittee X3J7 on APT Language Standards.

3. Insist on the use of the IPC-D-350A Standard on "End Product Description in Numeric Form for Printed Wiring Products" for future wedges relating to electronic systems.

NBS-114A (REV. 7-73)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 76-1094 (R)	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE STANDARDS FOR COMPUTER AIDED MANUFACTURING Fourth Interim Report			5. Publication Date April, 1977 6. Performing Organization Code 600.20
7. AUTHOR(S) Dr. John M. Evans, Jr., et. al.			8. Performing Organ. Report No.
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			10. Project/Task/Work Unit No. 6007404 11. Contract/Grant No.
12. Sponsoring Organization Name and Complete Address *(Street, City, State, ZIP)* Manufacturing Technology Division Air Force Materials Laboratory Wright-Patterson Air Force Base, Ohio 45433			13. Type of Report & Period Covered Interim 14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

This report identifies and evaluates those existing and potential standards which will be useful to the Air Force in the development and implementation of integrated computer aided manufacturing (ICAM) systems. Such systems, when implemented by the Air Force and by Air Force contractors, will increase productivity in discrete part batch manufacturing by several thousand percent. The use and importance of standards are considered in the context of CAM systems. Since the Air Force will develop the detailed ICAM architecture after this study is complete, existing system concepts and architectures are examined to identify the common elements to guide the further presentation and discussion of relevant standards.

17. KEY WORDS *(six to twelve entries; alphabetical order, capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

CAM architectures; computer aided manufacturing; computer systems; standards; system integration; voluntary standards.

18. AVAILABILITY	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PAGES
[ ] Unlimited [X] For Official Distribution. Do Not Release to NTIS	UNCLASSIFIED	
[ ] Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13 [ ] Order From National Technical Information Service (NTIS) Springfield, Virginia 22151	20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price

USCOMM-DC 29042-P74