SOFTWARE TOOLS
FOR
SHIPBUILDING PRODUCTIVITY

FINAL REPORT
PREPARED BY:

C. Pieroth
Grumman Aerospace Corp.
Bethpage, NY 11714

and

R. Skirkanich
Grumman Data Systems Corp.
Woodbury, NY 11797

December 1984

A project of
NATIONAL SHIPBUILDING RESEARCH PROGRAM

by

SOCIETY OF NAVAL ARCHITECTS AND MARINE ENGINEERS
SHIP PRODUCTION COMMITTEE PANEL SP-4
DESIGN / PRODUCTION INTEGRATION

Contract DTMA 91-82-C-20018
Newport News Shipbuilding POM-67800-R
Grumman Data Systems project AD-8349

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
DEC 1984	**N/A**	**-**

4. TITLE AND SUBTITLE
Software Tools for Shipbuilding Productivity

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Surface Warfare Center CD Code 2230 - Design Integration Tools Building 192 Room 128 9500 MacArthur Bldg Bethesda, MD 20817-5700

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**231**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

ACKNOWLEDGEMENTS

How to communicate and use scientific and technical information about the process of Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) effectively has been a problem of serious and growing concern to industry. The steadily expanding volume of data and emergence of new disciplines, as well as new links between existing ones, has increased the number and diversity of user groups and - needs. Consequently, this report, and any insight to the Design/Production integration process it might provide, owes its existence to the contributions of a very diverse group of peopIe. Mr. P. Wiedenhafer, of Grumman Aerospace, who initiated the "Design / Production" logo of SNAME Ship Production Committee PaneI SP-4 (the "/" remove the bar theme) at the January 1981 SP-4 meeting aided in the early stages of this study. Dr. D. Doucette, Mr. R. Pouder, and J. Howe of Grumman Data Systems made many direct contributions. Relevance to the ShipbuiIding/Design process would have been impossible without the National Shipbuilding Research Program Reports, especially those penned by Mr. L. Chirillo. Innumerable industrial and research reports outlining CAD/CAM application completely foreign. to shipbuilding form a baseline on which to build the CAD/CAM applications investigated. The "Survey of CAD/CAM Technology Applications in the U.S. Shipbuilding Industry" and personal collaboration of Mr. R. Diesslin of the Chicago-based IIT Research Institute also gave added depth to this effort.

Another source of substantial contribution was the SP-4 administration itself, the MarAd Review Groups, the CAD/CAM Advisory Panel, and cooperative efforts of the IIT Research Institute personnel performing a parallel CAD/CAM study. These groups served to enable presentation and interchange of research data which added much to both the formulation of the problems to be addressed by this study, as well as to the knowledge and insight to resolve problems posed.

A major concern of this study was to maintain effective contact with governmental agencies and other industries involved in CAD/CAM integration activities to effect a technological transfer of the important aspects of computer technology which have a bearing on future shipbuilding practices. This consideration was a key factor in formulating the approaches taken to carry out this study, and the contribution of this baseline of CAD/CAM technologies is gratefully acknowledged.

Lastly, but not least, are the first-hand contributions, supplied on site, in shipyards throughout the country by the over 180 personnel involved in shipyard and design agent operations who were interviewed during the report preparation period. These contributions are all gratefully acknowledged, and taken as a whole are probably the most unique mixture of contributors yet assembled to aid in the newly emerging, uncharted areas of shipbuilding design/production integration.

This report is the end product of a Project managed and cost shared by Newport News Shipbuilding for the National Shipbuilding Research Program. The Program is a cooperative effort of the Maritime Administration's Office of Advance Ship Development and Technology and the United States shipbuilding industry. Industry direction is provided via the Society of Naval Architects & Marine Engineers' Ship Production Committee.

CONTENTS

CONTENTS (Cont.)

Page

CONTENTS (Cont.)

CONTENTS (Cont.)

CONTENTS (Cont.)

Page

APPENDICES

LIST OF FIGURES

(Source citations far figures are summarized in Appendix F)

LIST OF FIGURES

1. EXECUTIVE SUMMARY

The objectives of this study are to define and identify software tools, and to impart to the shipbuilding community the knowledge to use them to aid in the design/production integration of the shipbuilding process. The approach taken in this study has been to:

- *Research, review and define the CAD/CAM integration process*
- *Develop selected scenarios of modern CAD/CALM integration methods*
- *Isolate and research software aspects of CAD/CAM scenarios; select and list application areas for software tools to (potentially) increase productivity for the integration process defined*
- *On-site visits to shipyards to review prepared CAD/CAM scenarios and software tools to define applicability of technologies, need for changes and knowledge-level of potential users*
- *Collect, reduce and review data from on-site visits*
- *Create a scenario adapted to the real-world of shipbuilding; identify critical software needs and select useful software tools*
- *Select a shipbuilding scenario and determine software needs to actually generate the integrated system; outline a means to calculate potential savings through the use of software tools.*

The material presented is ordered as outlined, and is followed by a catalog of software tools, and a recommended means of distributing results to the shipbuilding community. A glossary of acronyms is also included. There is no attempt made to specify currently in use, or projected hardware/software systems in either the computer, or CAD/CAM device arena. This task has been undertaken in approximately the same timeframe as this study by the CAD/CAM Survey Study performed by the Chicago-based IIT Research Institute which is reported separately.

This report has as its focus, the identification of CAD/CAM integration requirements and software tasks required to support them. The categorization of these software tasks into logical steps amenable to increased productivity by application of specific software tools is the end product of benefit to the shipbuilding community. Tools and the knowledge to use them, in this case for increased CAD/CAM software productivity in the shipbuilding design/production process, is the theme of this report. A broad recommendation to seek some standardization of the use of software tools to enable better Navy-industry, and intra-industry automated interface of integrated CAD/CAM systems is the fundamental conclusion of this report.

2. INTRODUCTION

As the need for consolidation and integration of the shipbuilding Design/Production function has grown, many new technical disciplines have been introduced to the ship design and shipbuilding communities. CAD/CAM systems (Computer-Aided Design/Computer-Aided Manufacturing), have emerged as a viable focus of these new technical disciplines. The large-scale exploitation of computer capabilities in this CAD/CAM context offers the possibility of increased refinements in tailoring new design/production processes to specific shipyard demands. However, large scale exploitation of the capabilities of computer technology to enhance shipbuilding design and production activities is dependent upon successful utilization of computer software, compatible with both the target computer, and the support environment afforded by each shipyard.

Software is not a clearly discernible focus, but is nonetheless a pivotal productivity and cost issue for the design/production integration process. The identification and use of software tools to aid support of the CAD/CAM integration process in shipbuilding is the focus of this study.

Software planning, requirements generation, specification, test-plan creation, coding, testing, debugging, validation/verification, and final documentation, installation and operation becomes the bottom line in making possible the effective use of computer devices. Since computers, associated. mainframes, communication equipment, and peripheral devices are requisites of a CAD/CAM system, the software which drives these systems is then the single most critical factor in the use of new computer technology in shipyards.

The use of advanced techniques for improving the efficacy of software design, development, and maintenance will greatly aid the installation of advanced CAD/CAM and related shipyard computer systems. Current computer software technology has matured to the point where software tools are available to augment the software process during the entire software life-cycle. The objective of this project shall be the assessment of planned software development needs to support

CAD/CAM, the identification of modern software tools available to economically facilitate software development, and the creation of a framework to assess costs and benefits accrued through the use of these tools. A tangential benefit of software tool use is the potential for enabling the sharing of the application software developed between yards. Unlike hardware, software can be shared between users, and software tools can allow this sharing to take place even if computer hardware is different.

An abbreviated statement of the prime objectives of this project are:

- *Develop scenarios of integrated shipbuilding **CAD/CAM** systems.*
- *Determine the current plans for computer software development, purchase, and maintenance in American shipyards to support hypothesized "integrated" systems.*
- *Outline a means to calculate cost, manpower/skill types, software lines of code estimates and ancillary, resources required to support these **CAD/CAM,** and related functions.*
- *Evaluate existing software tools commercially available to facilitate shipyard software systems development, production and test.*
- *Using a benefit measurement system approach, show how to determine the cost savings for application of selected automated software tools to typical development scenarios in shipyard **CAD/CAM** environments.*
- *Compile a report to enable selection, evaluation and use of software tools by interested shipyards.*
- *Propose a plan to disseminate the report to the shipbuilding industry.*

*At present, there is no single directory of information sources keyed to shipyard **CAD/CAM** needs on which to base selection of software tools. This project will provide an assessment of existing automated software tools that can be used in the* *development and integration of current and future software tasks. A means to estimate benefits using selected methods will also be included.*

Information was gathered by visits to shipyards where on-site interviews with staff members of software operations and many other departments were carried out. This information was then compared with the software development tools and experience currently in use in other manufacturing industries. The specific requirements of shipyard needs are used as the criteria to select candidate software tools. Descriptions, uses and the availability of selected tools are cataloged in a handbook format.

Specific technical approach areas investigated have included investigation of the following classes of software tools:

- ***System Requirements Generators.*** *Means of resolving system needs into a form suitable for specification generation.*

- ***Data Directory Systems.*** *Tools used to develop common data terms/usage for data base usage that insure an orderly software development process.*

- ***Conversion, Transition and Translators.*** *Methods used to enable efficient, consistent update of 'old code' to new machines and systems cost-effectively.*

- *Mediational Utility Methods. Methodologies and tools with the ability to create an automated means of linking selected software outputs to desired input formats of other programs without changing internal code structure.*

- ***Documentation Aids.*** *Systems that enable economical, rapid documentation of software in an automated and consistent manner.*

- ***Configuration Management.*** *A methodology to create a software environment with identifiable, controllable logic, that makes software change control and standardization practical.*

- ***Software Management Systems.*** *Software tools to create an automated means of enabling a cost-effective environment to control software development, test and maintenance.*

These technical approaches have been accomplished by carrying out the following specific tasks:

- *Survey of industry interest, need, and support for the project. This effort was the basis for the project start recommendation.*

- *Visit and interview of personnel on-site in selected shipyards; including manufacturing/design, production, planning, purchasing and software departments.*

- *Match projected software needs in CAD/CAM areas by categorizing shipyard needs and comparing with other industries to affect technology transfers.*

- *Assess software tools available and match these against established shipyard needs and software priority areas established by hypothesized shipyard applications.*

- *Create a report outlining software tool types, usage, availabilities and probable paybacks for selected shipyard CAD/CAM applications.*

- *Create a plan to communicate the results to the ship design/building industry.*

A Guide outlining project findings in the form of specific recommendations keyed to the report by chapter has aIso been prepared and included in the appendix. This Guide is suitable for use as both a teaching tool and as an introduction on how to use software tools in a shipyard software environment.

A shipbuilding industry briefing has been provided to familiarize the industry with software tools, the results of this project and the use of its deliverables. An abstract of this briefing is included in the appendix.

This report is designed to be used as a road map to plan a variety of general approaches to automating the shipbuilding design/production process, and to enable the specific application of software tools to augment the production of software to support the CAD/CAM process. Software tools and the knowledge to use them in the shipbuilding CAD/CAM integration process is the focus, purpose and intent of this report. It should be of value to the ship design, building and customer community subtlely through gaining increased productivity of computer functions required to integrate CAD/CAM functions, and overtly through bringing these functions on-line quicker and maintaining their efficiency throughout the years.

3. CAD/CAM INTEGRATION: THE PROMISE AND THE PROCESS

3.1 BASIC PRINCIPLES

A primary concern throughout this study was to relate the complex development of CAD and CAM Systems, both hardware and software aspects, to environments which have shaped emerging CAD/CAM Integration trends. Implicit in this task is the apriori definition of terms, including "CAD," "CAM" and "CAD/CAM Integration." Each of these terms must be defined and explained, in the context of the environment selected as most nearly applicable to current shipbuilding needs, thus providing a foundation on which to view sets of performance characteristics, useful processes and systems of value to the shipbuilding Design/Production Integration process. This philosophy enables the identification of viable CAD and CAM benefits, as well as the process by which CAD/CAM can be integrated into a Design/Production system. Once identified and defined, technology transfer techniques can be used to formulate approaches to affect CAD/CAM Integration of the shipbuilding process.

There are several reasons for this study approach. Apriori isolation of CAD, CAM and CAD/CAM technologies before studying the needs of shipyards enabled an unbiased data base to be developed containing real-world CAD/CAM systems and their applications. This enabled an analysis of both the promise (benefits) of these systems in a Design/Production environment, and the process by which they were (or were intended) to be integrated. This latter activity supported the thesis that the management of data, via colmputer techniques, is the central process that makes CAD/CAM integration practical. The need to increase software design and development activities to perform these data management tasks makes it necessary to increase the understanding of the software development process and software productivity issues to assure CAD/CAM integration. Applicability of software tools to the process of CAD/CAM integration and attendant software needs to facilitate this integration is the focus of this study. The early development of CAD/C.AM issues from the related industries enabled the study of the software process applicable to the development of Integrated CAD/CAM systems.

3.2 INTEGRATION RESULTS

Recent years have seen a proliferation of CAD and CAM systems applications in many industries. Many of these systems have purported to be totally integrated systems. For the manufacturing environments applicable to shipbuilding, none can be considered to be worthy of this classification. However, these many starts towards the automated factory have given three very valuable insights towards accomplishing the goal of the totally integrated Design/Production process:

- *Islands of automation are not necessarily desirable*
- *The Design/Production process, as it trends towards computer orchestrated integration, has become more complex than the product being manufactured.*
- *Managerial, organizational and personnel factors rank equally with technological factors as impediments to CAD/CAM integration.*

The promising productivity performance of those selected sectors of both the CAD and CAM process which have been automated, creating islands of automation, are often obscured by the complexities of attempting to integrate and control the CAD/CAM process as a whole. This report section will deal with the developments to date which have yielded CAD/CAM productivity improvements, and describe the process of attempting to combine the "Islands of Automation" into a data-driven system representing "Design/Production Integration."

3.3 DISCRETE BATCH PRODUCTION. THE CAD/CAM ENVIRONMENT

Computers have been used in the industrial environment virtualIy since the inception of the computer over three decades ago. However, the type of industrial environments they were first successfully used in were able to accept much less computer sophistication. A general categorization of the three types of manufacturing industry environments are:

- *<u>Continuous-Flow Processing</u>: Characterized by control of valves, thermostats, and mixing processes to accomplsih a manufacturing objective; such as in petrochemical plants.*

- o *<u>Discrete Mass Production</u>: Characterized by making thousands, or even millions, of repetitive products; such as bottles, tin cans or cigarettes.*

- o *<u>Discrete Batch Manufacturing:</u> Characterized by design/manufacture of many complex parts in small lots, which in turn, must be assembled into larger systems, of which the total number of final products are relatively few; such as aircraft, machine tools, and ships.*

The Discrete Batch Manufacturing environment characterizes the majority of manufacturing tasks in the United States. It is the most difficult of the manufacturing environments to automate because it requires frequent (sometimes constant) changing of tooling, machinery, product design, materials and assembly sequences. The many individual operations required, complicated by the need to share available equipments with a day-to-day change in the mix of products caused by the vagaries of shop scheduling, product orders and inventory availability, demands a great deaI of manipulation of personnel assignments and allocated facility space as well as machine tool usage.

Additionally, the discrete batch manufacturing environment must deal with the often daily changes in the end product configuration demanded by the customer for legitmate technological requirements, design change necessity or sheer whimsy. This last point, the need for constant handling of changes, underscores the need to integrate CAD, CAM and MIS (Management Information Systems) in Design/Production Integration of the discrete batch manufacturing environment.

The transfer of CAD/CAM technologies and equipments from the discrete batch Manufacturing environments to shipbuilding is the investigatory approach taken in this study. This tact is selected because shipbuilding is an example of discrete batch manufacturing. The following discussion of CAD and CAM concepts are in the context of the myriad of problems encountered in the discrete batch manufacturing environment.

3.4 COMPUTER AIDED DESIGN (CAD)

Technological advancements in Computer Aided Design (CAD) Systems have evolved with the growing sophistication of computer hardware making possible low cost, mass storage media. Many current CAD Systems have small-scale mini or micro computers as an integral part of each display screen to enable rapid-response to user needs without placing a heavy burden on computer resources. This arrangement also enables a great variety of user friendly graphic interaction and high resolution graphic output, made possible by a wide variety of modern display technologies.

Notwithstanding these advancements, as a standalone system, modern CAD applications can be of little technicaI advantage, in terms of increased productivity, over systems available years ago. This restricted productivity potential may occur when existing Design/Drafting departments are updated by the insertion of a CAD system into an existing organization without due consideration of the potential for CAD to be integrated with other processes to aid a company's strategic mission. Initial improvement in the output of the Design/Drafting area may soon amortize the costs of the CAD equipment, and even encourage expensive expansion/updating inducing acquisition of similar CAD equipment. However, this may only create what has become known as the classic "Island of Automation" wherein a small sector of operations is visibly automated, without the ability to beneficially improve the pace of surrounding Design/Production areas. Taking full advantage of the growing areas of CAD potential can increase the productivity of many other areas beyond classic CAD system capabilities. These classic capabilities, and their uses, are described in this section. The capabilities described are limited to a "stand-alone" CAD system concept to gave a base line of understanding; to the advantage of CAD before consideration of CAD/CAM integration in later sections.

3.4.1 CAD System Components

Computer Aided Design (CAD) systems have evolved into practical productivity aids only during the last few years. There are easily defineable parameters on what constitutes a typical CAD system, any which typical system is composed of *"common (CAD) system elements". However, CAD systems for any specific in-house system must be explained in terms of three different descriptions. These descriptions are:*

- *o* *System Development Methods: How the constituent CAD elements are introduced, and integrated, to perform the function for which they are selected. These approaches include time sharing networks, turnkey systems, integrated systems and internally developed systems.*

- *o* *System Applications: What functions the CAD systems are designed to perform are predestined by their innate capabilities. CAD system types can be multi-function, special purpose, or mechanical design/drafting oriented as to function.*

- *o* *Common System Elements: Composition of computer graphics systems is made up of hardware elements, with accompanying software to tailor these to the selected system development methods and types of system, which include the following: computer, data storage media, CAD workstation (usually a Cathode-Ray Tube capable of user interaction via light pen or track ball; Alpha Numeric keyboard and function buttons), hardcopy output device (drum, flatbed, or electrostatic), auxillary equipment for drawing input (manual or semi-automatic digitizer).*

A schematic diagram of these CAD system components and their described interrelation is shown in Figure 3.4.1: CAD system elements

The Figure 3.4.1 diagram shows that the same common CAD system elements (hardware components) can be utilized for a variety of different types of system applications, and can be "planned" to grow by a variety of different system development methods. The difference is the type of software and planned capability to use this software over the life of the system. Subsequent sections will deal with these different approaches to CAD system use and growth, and highlight where integration of functions with other systems is required.

CAD SYSTEM ELEMENTS

SHOWN IS A TYPICAL CONFIGURATION OF CAD SYSTEM COMPONENTS, NOTE THAT COMPUTER SOFTWARE IS REQUIRED TO CONTROL FUNCTIONS', COMMUNICATIONS, GRAPHICS AND CALCULATIONS,

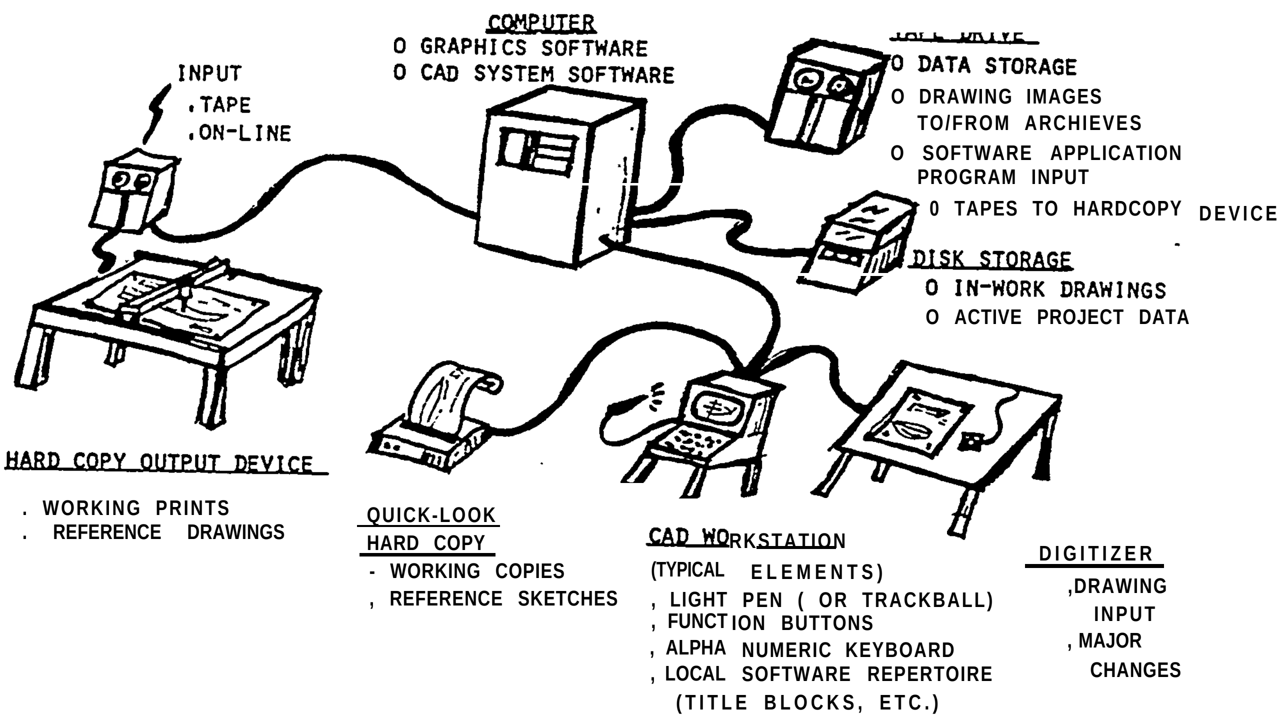


FIGURE 3.4.1: CADSYSTEM ELEMENTS

3.4.2 CAD System Benefits

Historically, CAD Systems have virtually all been, and many still are, nothing more than computer driven graphics systems which only serve to automate drafting functions. This is not to say that these CAD Systems are without merit. On the contrary, they are important to both. cost savings and Productivity/Quality imporvement in the drafting room. The point to be made is that often these systems have little to do with increasing actual design capabilities, and less to do with integration with other Design/Production functions.

Nevertheless, CAD has benefitted design automation in many measurable ways, even within this limited usage. CAD greatly reduces the time it takes to do an initial drawing. This benefit is accompanied by the bonus of an increase in accuracy, accrued with another saving, that of a lessened need for print checking. Since all drawings on the CAD System are entered onto a central database, the reliability of drawings is increased because the latest modifications are included on all output graphics.

Designers and/or Naval Architects can work with a CAD System by utilizing rough sketches as initial input, and using available input mechanisms to input these to the CAD System. A key to productivity improvement is the ability of a CAD System to capture a drawing detail in memory, and then reproduce it instantly at any desired point on the drawing. This avoids repetitive sequences of drawing, and cuts down on input effort. Approved design drawings can be accessed by detailers who utilize the CAD System to input additional information layers in the form of piping, electrical, structural, ventilation, communication and other specialties. These can be output in the form of combined system review prints, and even inked in different colors to provide a means to review for interferences. Scale changes can be accommodated quickly, as can changes to drawing details.

Importantly, CAD Systems insure that all drawings are of uniform graphics standards and reproduction quality. Output graphics are clear and concise, independent of the skill and dextrity of the operating designer and/or draftsman.

This ability of the designer and draftsman to collaborate on a single output drawing is an overlooked, but actual benefit of a first time through drawing using CAD Systems. This enables some justification to the claim of productivity improvements of 300 to 700 percent claimed for installation of CAD into a Design Department.

CAD enables effective exploration of more design options, and the consideration of a greater number of new items and tests of their consequences. The ability to do computer calculations virtually eliminates the time and errors involved with mathematical calculations.

Review and use of drawings by different departments is facilitated, as is the degree of accuracy for ordering material for manufacturing. Though many newer systems have stressed a quantitative improvement in. upgrading of their CAD Systems in the past, such as a greater number of terminals and faster drawing production, the newer trend toward qualitative improvements is more important. Qualitative CAD improvements yield the ability to do new design tasks, and thus affect increases in creativity. An important by-product of this trend is the growing ability of CAD Systems to break out of the drafting room and begin integration with other Design/Production functions.

3.4.3 New and Future CAD Technologies

CAD technological advancements have been announced, at an increasing rate with the advent of cheap large capacity computer accessible memories and more advanced software. Generally, this growth in sophistication can be characterized as a progression from the 2 dimensional (2D) CAD System, representative of early "automated drafting" devices, through 2½ dimensional (2½D) systems allowing notation of "Z" dimensions, through current solid Modeling (SM) Systems. SM-CAD is a System that enables user input references to a point or points outside the plane forming the surface of a graphically represented object. Many hardware manufacturers lay claim to solid modeling capability, with many other software firms claiming total SM systems capability. An interesting point of contention centers *on the claims of hardware vendors to be able to completely utilize SM independently of software. A contrary claim is made by software vendors who claim their capabilities are the most important considerations, with the display device and computer selection remaining secondary or tertiary considerations.*

Approaches to storage of data for display are divided into two major camps. One method is to maintain an exact mathematical model, while a second approach maintains an approximate relationship of bounded surfaces, or facets. Two schools have emerged on output display presentation of the stored data. There are advocates of Wire Frame Displays and Color-augmented smooth-shaded outputs. Many operational features for manipulating the solids, such as rotational capabilities to view shapes from selected viewing points and the ability to enable user interaction on a geometric component of a shape for later reassembly, are CAD-SM features.

All of these points of contention are shaping the CAD technology of the future. Many of the basic competing issues have been debated academically since their introduction in the late '60's and early 70's, but are only now involved in heated public debate due to the increasing availability of CAD-SM at the industrial level. New hardware, software and display technology have made these technologies, if not cheap, at least marginally affordable. A second factor is the increasing availability of workable European developed CAD-S.M systems, and this is evidently creating a market impetus for general acceptance of advanced CAD.

Though an over simplification of the problem, the focus of the change from 2D to full 3D CAD-SM systems is the critical software/Data Storage process of orderly computing and output of an objects Z axis component. The CAD-SM Systems being developed resolve this problem by using the math models approach, faceted approach or combinations of these two" approaches. In conjunction with these software approaches, a broad spectrum of combinations of hardware, software and dispIay technologies are utilized. CAD Systems, of all descriptions, are here to stay, and of great importance to specific productivity issues. Their emerging role as a part of an integrated production system is just beginning to evolve. These emerging issues will be developed, using the survey presented here as a base, in later chapters of this report.

3.5 COMPUTER AIDED MANUFACTURING (CAM)

Profound changes in the American manufacturing environment began to materialize in the late 1960's when computers made their debut on the shop floor. The use of computers in manufacturing to improve productivity and efficiency was called Computer Aided Manufacturing, or "CAM". A working definition of CAM is the "effective utilization of computer technology in the management, control and operations of a manufacturing facility through either direct or indirect computer interface with the physical and human resources of a company." CAM evolution has taken a different course of development then CAD. An understanding of CAM rudiments and roots is important to the grasping of CAD/CAM integration concepts.

3.5.1 CAM: Historical Development

Although an intuitive assessment at the present, the idea of CAM accessing output generated from a CAD System and automatically controlling a set of machining processors to effect the creation of a finished part was not a part of CAM development history. CAM evolved as an offshoot of Business ADP functions in the late 1960's. The starting point of a CAM program was usually a set of points, either manually prepared or drafted by an output CAD program. In either case, the goal was the preparation of a punched paper or mylar tape containing instructions to implement the creation of a part by a milling machine, flame-cutter, or other Numerical Control (NC) driven device. NC is a system in which machine tool actions are controlled directly by entered numerical data, where the system automatically interprets some of the entered data. Machines were thus automatically operated via input of discrete numerical values read from machine interpretation of data stored on punched paper/mylar tape, magnetic tape, or direct computer control.

Frequently, NC machines were misunderstood by manufacturing executives. The basic misunderstanding, persisting through the present, is that an NC device is recommended only for large lot sizes of the same part. Contrarily, NC is ideally oriented to small size batch jobs. The ability of NC devices to store both cutter operation information, and tool selection instructions, optimizes the ability to work *on a large number of different parts with a minimum of set-up time. Tools are automatically selected from a carousel or similar machine controlled tool nolding device. Even lot sizes of one are economically feasible in many cases. Surprisingly, NC is not viable for extremely large quantities - special purpose machines usually give better economies for long lasting runs of large quantities.*

Numerical control does not influence machining characteristics, or as originally conceived, cut faster than good shop practice for the materials/cutter/machine combination being used. CAM use of NC has the great advantage of keeping a machine "busy" (cutting) a greater percentage of the time, thus raising average productivity. However, over 75 per cent of a machine tool's time is spent in various activities other than actual productive cutting operations, even using conventional NC machining operations. These include set up, waiting for materials and other unproductive activities. Clearly, the ability to integrate work processes would allow considerable additional increases in productivity.

A brief overview of NC advantages are listed as follows:

- *Improved Reaction Time: A part that is on NC tape can be made in a matter of hours, as compared to days for a part from prints. This aids "Queue-Time changes on the shop floor for priority tasks.*
- *Accuracy: Tolerances are held to closer limits, and the mating of parts is facilitated since all are within the designed tolerance.*
- *Operation Experience: Formal training and lengthy experience is not required; a trained operator can replace a skilled machinist, with the same output results.*
- *Scheduling: Tighter schedules are held with NC manufacturing cycles, a great CAM aid to shop management.*
- *Inventory: Large stores of inventory items are not required, as the ability to economically make another "Batch" of parts when needed is. afforded by the speed of using the same NC tape. This conserves capital, storage space, warehousing costs and stock shrinkage.*

o Floor Space: Less space is required for an NC device when compared with a mix of machines to effect the same output of parts using conventional methods.

o Scrap: Proven NC tapes will virtually eliminate scrap; this enables less raw stock inventory to be held on hand.

Programming for NC operation can be done manually, even for quite complex parts. A computer is a great aid in this process. However, the starting point has always been a manually interpreted drawing, even if the drawing it self was originally produced on a CAD computer system.

To effect a manual NC program for a simple two axis continuous task, a "planning chart" describing the various operations of the NC adapted machine tool is used. Each operation has an assigned numerical code which can be encoded onto a punched mylar or paper tape. Such tapes can be prepared using a simple teletypewritter system.

Programming can also be effected using an NC computer language. The most common program ming language has historically been "APT (Automatically Programmed Tool). Use of APT permits five-axis contours to be programmed for cutting on an appropriately complex machine.

CAM functions were directed at other productivity areas in its early days, aside from the noteworthy focus on NC. Other areas of CAM computer use included:

o Control of Machine Tools

o Test and Inspection

o Quality Control

0 Assembly sequencing

o Material control and handling

o Measurement of special parameters

o Plant monitoring in support of manufacturing.

Use of computers in the above ways enables employment of CAM to effect cost sarings in the following ways:

- *Material cost: redesign to utilize less or cheaper material; less scrapage and rejected parts.*
- *Inventory cost: Both stock and in-float inventory can be reduced.*
- *Direct Labor cost: Automation reduces these costs.*
- *Machine Utilization Cost: Use of machines a greater percent of time enables use of fewer machines to accomplish the same output per unit of time.*
- *,4ssembly Cost: CAM enables savings through ease of assembly due to matching part tolerances. Actual assembly of parts is a goal sought after since CAM'S beginning, but has remained an elusive goal.*
- *Material Handling Cost: CAM provides a means to enable on schedule flow and requisition of material.*

3.5.2 CAM - Summary

As the main thrust of CAM initially directed itself onto the shop floor alone, early productivity success succeeded only in isolating CAM from the Business and Engineer/Designers world. The Business ADP personnel of industry looked with some degree of fear at the success of the shop-floor processes, while the perfection of the ultimate in output print quaIity sufficed for the end goal of CAD. Succeeding sections will deal with the use of CAM in an integrated context to aid the shipbuilding environment.

3.6 CAD/CAM INTEGRATION

Conflicting views exist on the proper division of task responsibility between CAD and CAM activities when they are upgraded, required to communicate, or other. wise brought to a confrontation on a hardware, software or person-to-person level. The principal cause of this controversy is that both CAD and CAM system vendors are centering developments on their systems, expanding capabilities and the scope of their systems within the Design/Production cycle. This competition-driven quest for improvement has fulfilled a broad range of useful functions to make more efficient the individual CAD or CAM island of automation. However, notwithstanding the enhanced efficiency introduced into new systems, this practice has thwarted CAD/CAM integration activities in two ways throughout American industry. Firstly, it has made early productivity gains in CAD, for instance, refortify themselves by enabling purchase of "upgraded" systems devices by the same departments. This "bottoms-up" acquisition of capital equipment has served to solidify the walls of isolation around "Islands of Automation" in design, drafting and shop floor sectors. Secondly, new systems in both CAD and CAM arenas are packaged as seemingly efficient "standalone" equipment, enabling the same "bottoms-up" analysis to result in the purchase of several different, difficult to integrate systems.

These practices have enabled early productivity gains, recognized in different CAD and CAM systems to become eventual impediments to the integration of CAD/CAM systems. In fact, those very advantages stressed in earlier sections of this report for CAD and CAM are causing a stagnation of information exchange in many plants due to the advocacy of system specific features, many of which are now outmoded.

The vital importance of developments in the CAD/CAM integration arena to our country's industry in general, and shipbuilding in particular, cannot be overemphasized. This concept of integration presents many difficult new problems, not the least of which is the requirement to take a fresh, new look at the organizational *settings of CAD and CAM as they now exist, as well as the new technologies. Both established organizational precepts, as well as the earlier mentioned technological advantages of CAD and CAM will be attacked by the modern and future definition of integrated CAD/CAM concepts.*

3.6.1 Integration vs Synergism

Conventional approaches to CAD/CAM integration have limited approaches to better, more efficient means of having CAD and CAM systems "communicate" with each other. Importantly, this approach also extends to the need for CAD-to-CAD and CAM-to-CAM and CAM-to-CAD interfaces (see Figure 3.6. 1)

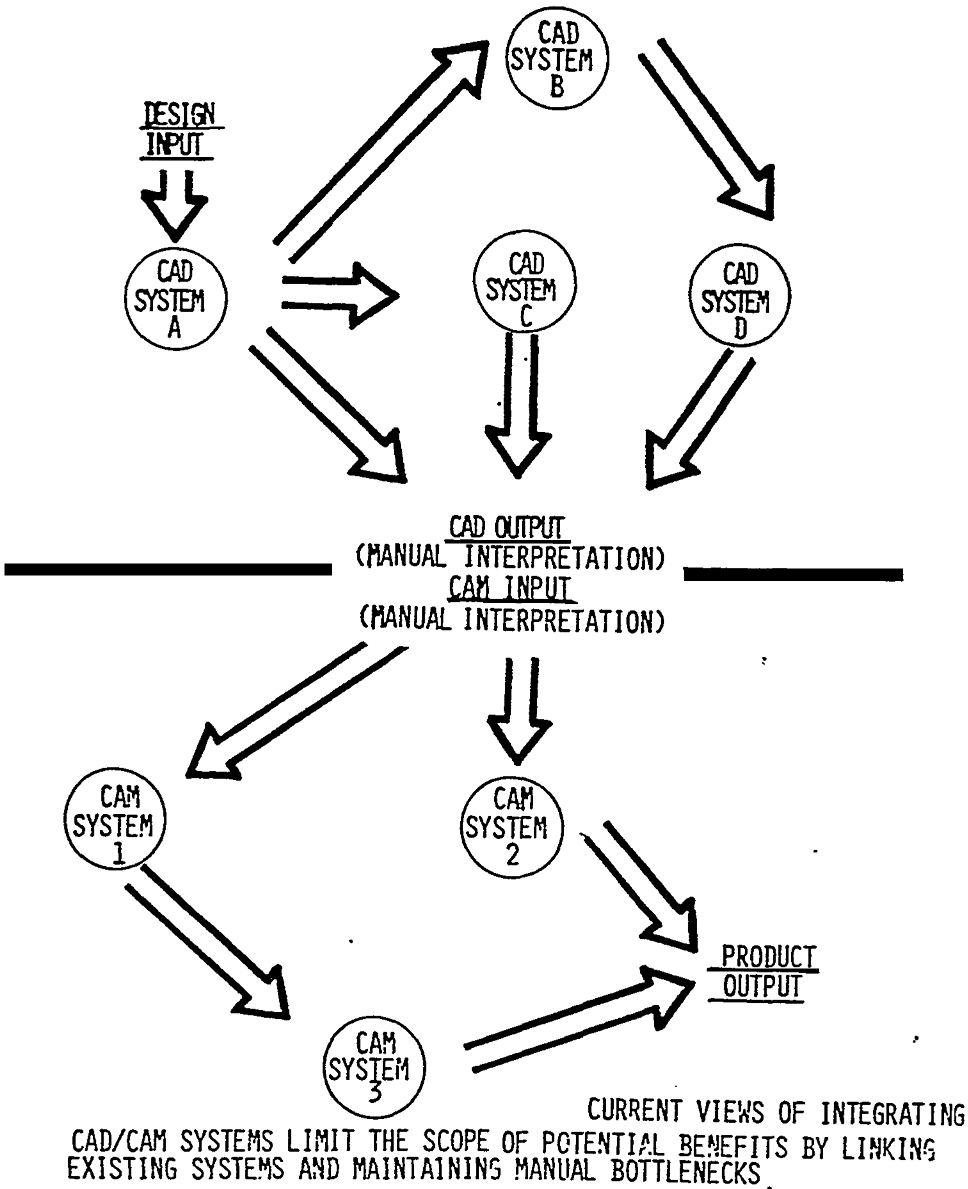


Figure 3.6.1: CAD/CAM Interface Matrix

However proper this approach seems, it makes a salient point with regard to good systems engineering practice: the sum of a system's components do not equate to a useful system. The CAD/CAM interface is unlike a simplistic union of the components via a phone line. It must be treated as a conceptualization of an entirely new system based on the requirements dictated by the environment into which it will be fitted. A singularly pervasive conclusion of the process of CAD/CAM integration is that the designing, consolidating, simplifying, and repackaging of CAD/CAM functions for a specific category of industrial applications must be thoughtfully planned and executed by a new class of system engineers. Synergism holds that the sum activity output of a balanced system is greater than the additive sum of its individual component subsystems. Likewise, the practice of creating a synergistic system often requires combinations of system components outside the initial list of parts from which to create a system.

CAD/CAM integration exhibits this quality of, and is created by this process of, dynamic synergistic system synthesis. CAD/CAM synergism is thus a more descriptive term for the combination of design and production functions. Examples of CAD/CAM synergy will be used as a basis to investigate shipbuilding CAD/CAM integration needs, and to create a working definition of CAD/CAM integration.

3.6.2 CAD/CAM Synergy: Integration Defined

Integration of CAD and CAM processes affect all of the business and technical interests of a company, and are of concern to so many currently unrelated disciplines and organizations that a dominant influence cannot be exerted by any existing group. However, efforts to address and resolve all aspects of the integration problem must-attend to all phases of company operation.

The view that is adopted in this report is that an analysis of the CAD/CAM integration process resolves itself into a definition of integration as CAD/CAM synergy, which is the fusion of selected CAD and CAM functions with functions of other business and technical areas affected by data created by and during these *functions in order that computer control of data. can effect orderly design-production processes at increased leveIs of productivity: Thus, the by-product of CAD/CAM synergy, data, enables the control of the CAD/CAM process as we know the individual CAD/CAM function, but the tangential areas affected, which are different for each industrial environment are also drawn into this integration process. As each discrete component of this process is only identifiable after considerable study, and the working interrelationship dependent on the data each process generates, the only common elements are the extraction of data itself. Data becomes the key driver to move an organization toward automation via computer manipulation of data flow by software control.*

This study will focus on such synergistic approaches to CAD/CAM integration, the needs of such a synergy in the shipyard environment, the effort to cope with the software tasks evolving from the need to effect orderly control of this data and thus the integration process. Identification of software tools as a means to cope with an orderly identification of CAD/CAM integration requirements, linking of diverse disciplines, and planning for future expansion of the integration process will be the focus of this report.

Figure 3.6.2 shows the overlap of the design-production data flow - a simplified schematic.

This diagram highlights the overlap of the process, which is different in concept from the older idea, still useful in many applications, of building data “bridges” between systems. By creating data bases, the computer architecture thus established enables inclusion of many more functions within the scope of the CAD/CAM synergy equation. Figure 3.6.2-1 shows an example of how this would evolve.

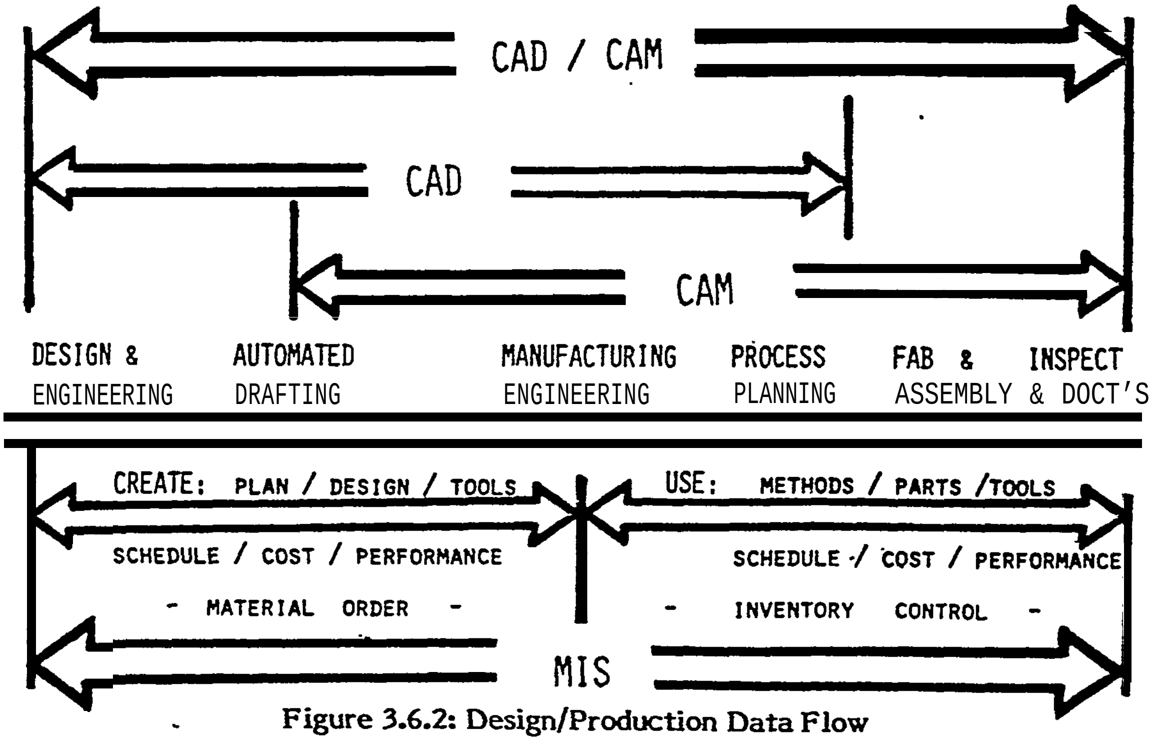


Figure 3.6.2: Design/Production Data Flow

Note how the ability to interface with diverse disciplines is both made, possible, and, in fact, required by this integration process. Importantly, the data bases, their communication with the system and the involved software become pivotal issues for the integration process. CAD and CAM functions, earlier covered, are relegated to equal, but not paramount, status in this structure. CAD/CAM synergy is a process of automation by data dynamics. The role of computer software becomes the singIe most critical issue to both effect this synergy and adapt it profitably to the individual shipyard environment.

The next two sections will deal with recognized strategies of CAD/CAM integration, and the software requirements to implement these strategies.

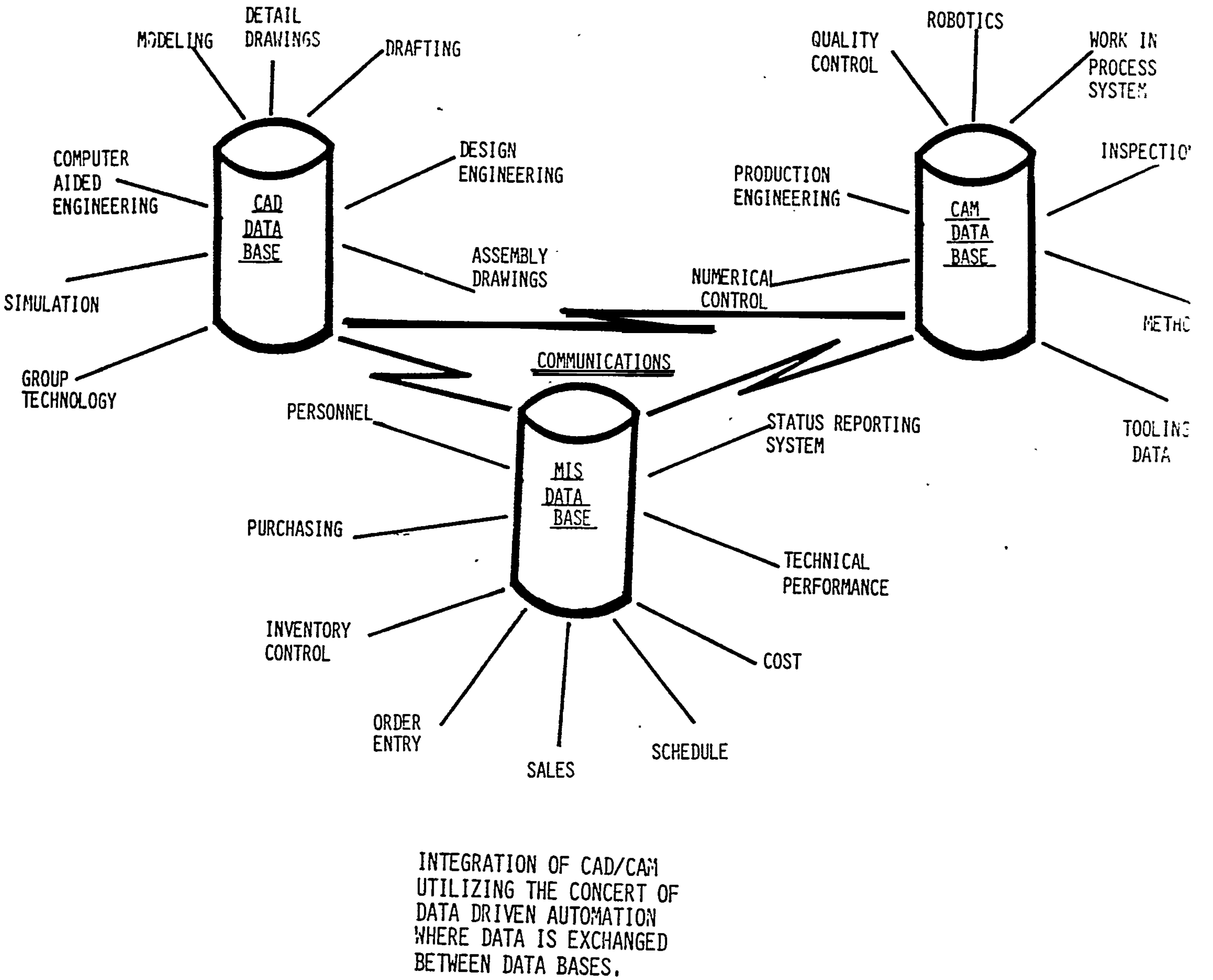


Figure 3.6.2-1: CAD/CAM Synergy Computer Architecture

4.0 APPROACHES TO CAD/CAM INTEGRATION: SELECTED SCENARIOS

This study is directed at detailing the use of software tools as a valuable aid to the CAD/CAM integration process in the shipyard of the near future. Thus, the investigatory process involved necessitates the forecasts of technology in the fields of CAD, CAM, Computer Science, and shipbuilding. Technological Forecasting (TF) is the identification of threats and opportunities in a companies future environment, and can be accomplished by many means. TF is thus a process of speculation on future useful applications of science, technology, and technique improvements and transfer from one field to another. This report uses forecasting to enable meaningful incorporation of integrated CAD/CAkJ into the planning of shipyard modernization and isolate aspects of software productivity to which modern software tools can be applied. Forecasting and planning are reIated, but not synonomous, and the mission of each is important to their effective use. The technological forecast has as its mission the presentation of facts in a manner that makes the formulation of a plan possible. Any forecast, technical or otherwise, is void of meaning without resultant actions, which actions are made possible by a plan. Thus planning has causitive and subjective attributes, where a forecast has no directed intent, and strives to be completely objective. .More specifically, .a technical forecast is becomming important to management to understand the basics of the requirements for new systems. The synthesis of these requirements from the environment and the subsequent analysis of requirements is the soundest means of deriving hardware and software requirements for integrated CAD/CAIM systems.

Key technical planning objectives, which can be achieved through use of technological forecasts, include the following

- *Identify: New business trends, threats and opportunities*
- *Relate: Identified trends, threats opportunities with corporate objectives*
- *Define: Technical requireinents associated with selected objectives*
- *Detect: Voids in existing technology*

- o Analyze: Alternate courses of action needed to reach technical goals
- o Specify: Realistic goals for technical programs within corporate capabilities
- o Evaluate: Alternate goals and determined priorities
- o Develop: Concise research and development program.

Forecasting, which forms the basis for achieving a viable plan, can be accomplished by a wide variety of developed TF methods. Our discussion will be limited to the mode TF has been employed in this study and not digress into TF techniques and theory.

There are two basic approaches to forecasting; Normative and Exploratory. Figure 4-1 outlines a conceptual schematic depicting the differences between these two forecasting approaches.

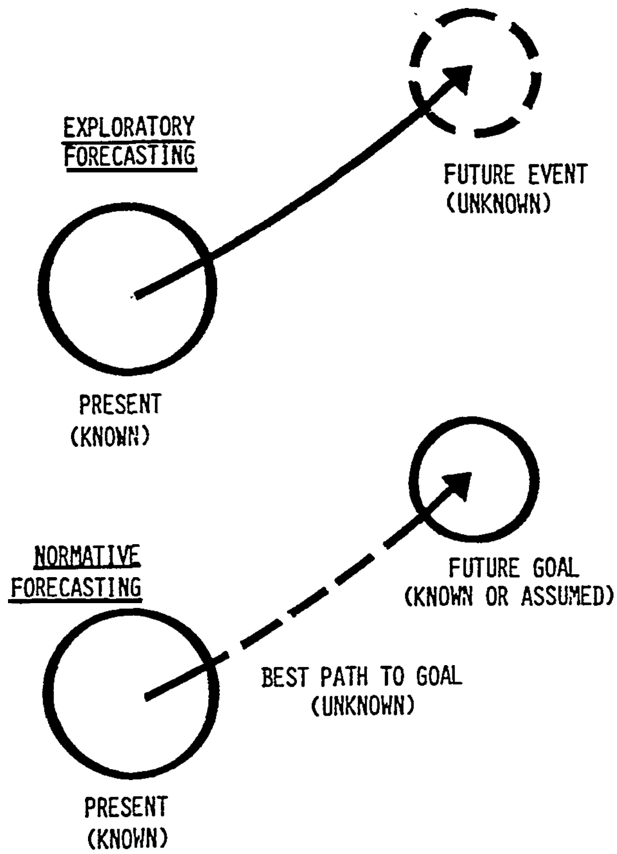


Figure 4.1: Exploratory and Normative Forecasting Approaches

Normative Forecasting is essentially a resource allocation method where the best technical means to attain a stated, or assumed, goal is sought. Normative forecasting often takes place as a part of a companies operational planning activities and becomes a part of the planning process itself. TF systems for ranking and evaluating research projects in selected areas involve normative forecasts of technology. When the goal of a normative forecast is specified, it becomes a part of a working plan. If this goal is undefined, or unknown, special attention by forecasters is directed towards it. The Normative Approach holds that this development is molded by interactions with the environment.

A contrast to the relatively structured approach of Normative Forecasting is the more conventional, much less constrained approach called Exploratory Technological Forecasting. This approach holds that what is possible to do, given a set of starting technological levels, will be done. A degree of limitation to this is set by simply imposing some degree of economic limitations, but aside from this, technological trend extrapolation dictates what state the investigated technologies will be in at some future time. A Normative approach to forecasting will focus on a chosen (known or assumed) goal and investigate alternate paths to identify an optimal method to reach the selected. goal Essentially this process determines a hypothetical future state and works backwards. An exploratory approach selects, from the known technolgoical base, most probable events and builds towards the potentially achievable technology given a time in the future and some degree of financial restraints. The two approaches can be linked by using an exploratory approach to define goals, and then using Normative approaches to outline ways to reach this selected goal

Viable technological forecasts cannot be obtained, whether normative or exploratory, in the absense of knowledge of the environment to give predictions some degree of context. TF actually begins with a hypothesized future environment surrounding the technology being investigated. Frequently, it is impossible to select an exact, or even an approximate, environment that will exist in the future.

To minimize the deleterious impact of misjudging future environments, a number of potential alternative futures can be examined. This examination can be carried out by the use of scenarios, which are alternative descriptions of what the future might look like, based on information gathered and analyzed by other techniques. In this study, the use of scenarios was selected to carry out this research program into software tools. Formal investigatory activities were aimed at refining CAD/CAM integration scenarios obtained from other industries, and the postulated future shipyard environment by use of both exploratory and normative forecasting methods. Effort using exploratory forecasting techniques was focused on the importance of software and software productivity to CAD/CAM integration. This issue was always examined after queries on the future of CAD/CA.M integration in the shipyard environment were discussed, in order to validate the future environment. The shipbuilding environment focus was augmented by having a professional Naval Architect/Engineer, knowledgeable in shipbuilding, as a participant at on-site visits to participating shipyards. This insured that relevance and importance were judged, avoiding an over emphasis on high visibility issues which may have little impact on shipbuilding.

4.1 SCENARIO DERIVATION AND TECHNOLOGY TRANSFER

Outlining a key area of productivity improvement in the shipbuilding Design-Production integration process requires a means to act on the forecasts made. This process is the Technology Transfer Process. An over simplified explanation is that a relevant science is input to a forecast; a Technology Transfer Process (TTP) is then employed to adapt the technology to the shipbuilding environment. Figure 4.1-1 outlines this process. The goal of the TTP is to assess the transfer of innovations and improvements from one field to another in order to determine the probability and time of occurrence of such transfers.

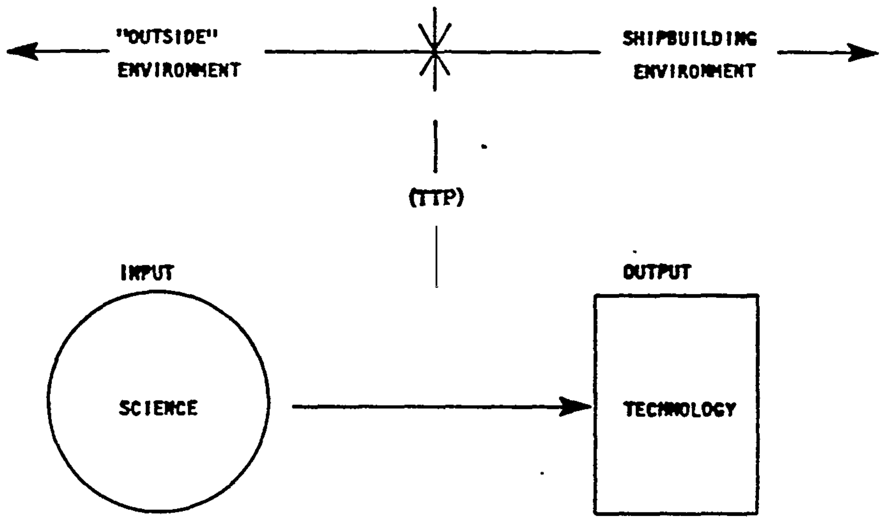


Figure 4.1-1: Technology Transfer Process (TTP)

Further breakout of this process identifies the components of the TTP. Reference Figure 4.1-2.

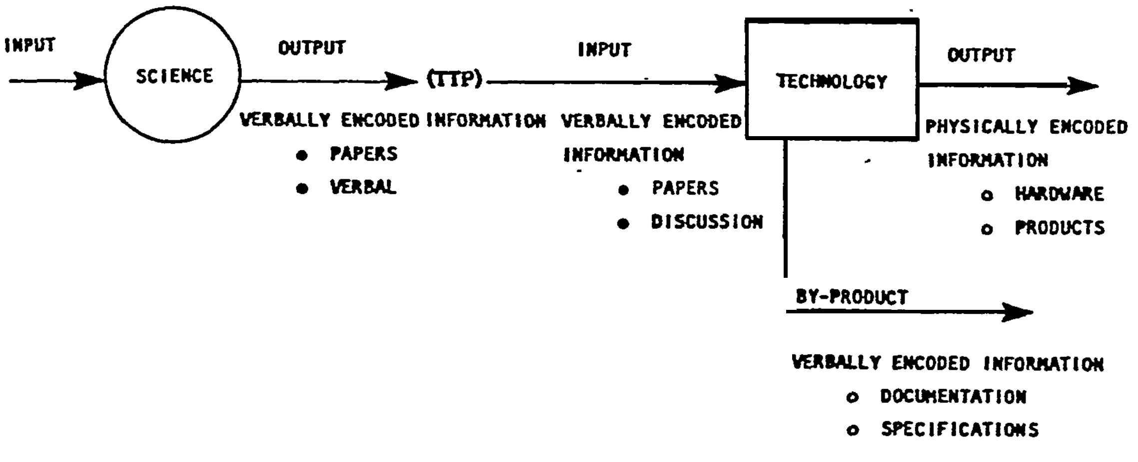


Figure 4.1-2 Components of Technology Transfer Process

The input for a forecast may, in fact, be a new specified technology, a technological concept, or a new technique. The adapation of these forecasts, via a technological transfer process, can be used to formulate a postulated view of a future environment in an industry. Knowing the hypothesized future environment, salient features can be isolated as potential goals for achievement and a "best path" to goals investigated via the normative forecasting technique. Figure 4.1-3 shows the manner in which the preparation of CAD/CAM integration forecasts were integrated into the future shipbuilding environment and projected into a format amenable to analysis of software took effects on the computer support required.

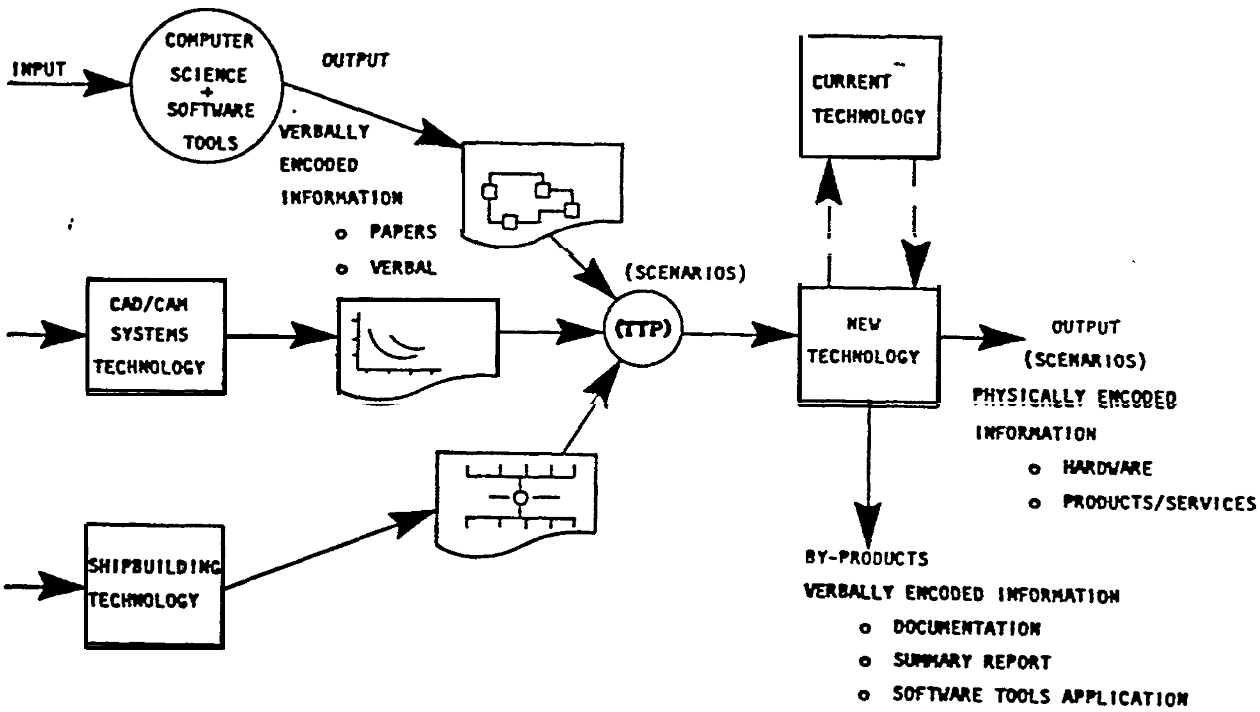


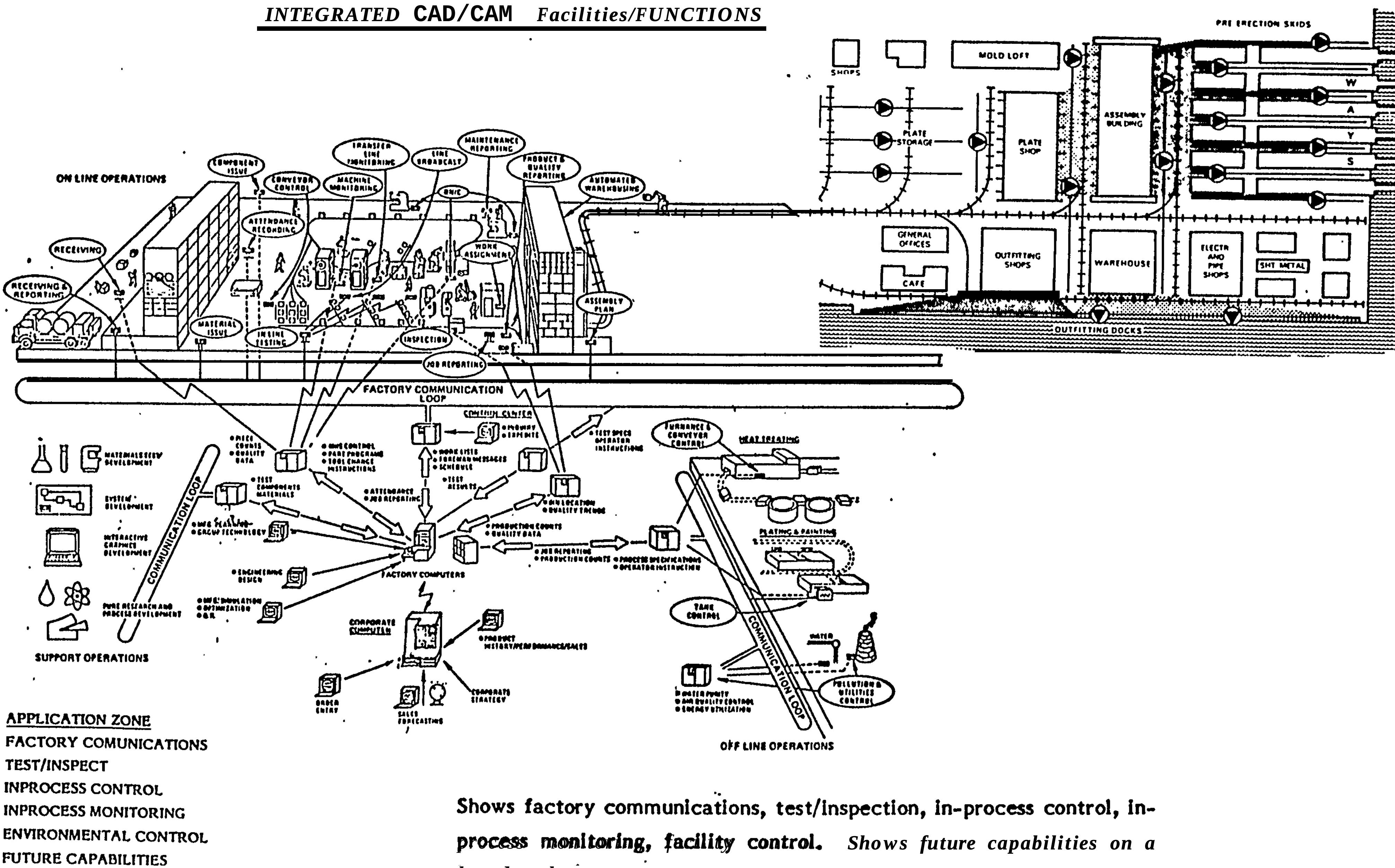
Figure 4.1-3: Use of Scenarious for Technology Transfer in Shipbuilding

4.2 CAD/CAM SYNERGY SCENARIOS

Scenarios depicting concepts and ccepted schemas of integrated design/production methodologies were sought in the literature, and classified by several categories. Selection of a series of these schemas was made, with the selection guided by two factors: (1) potential relevance to shipbuilding (the decision criteria was guided by discrete batch manufacturing applicability and basic fit to data found in notes from early meetings during the formation of panel SP-4; (2) representation of, and interlinkng to, major concepts projected in the CAD/CAM industry. The scenarios selected were resolved to clear, simple schematics. This use of graphics enabled ease of group review of scenarios and simple addition of concepts through pictorial notations, As all ideas generated were captured in preformatted scenarios, it was possible to categorize responses and relate comments on one scenario to others since their differences and similarities were known apriori. A tangential benefit of this scenario approach was that the proprietary nature of existing and future CAD/CA.M plans in the shipyards were protected, because generic approaches were used in all cases. This enabled commentary on technologies, in the context of the scenarios presented, which were all prepublished. Thus no compromise of specific computer hardware and/or soft-ware techniques was required. The output report is also generic in nature. This is of value because it enables use by a wide range of shipyards, since, by its nature, the output will be of general applicability and phrased in industry-common terms. No mention of vendors, software-hardware, or machine tools are made by name in the reports or scenarios, except if a scenario source was referenced to a particular vendor, an apriori fact of publication. In fact, only in the later section of software tools is any commercial data mentioned at all. A list of the scenarios used to elicit information during shipyard visits is presented below.

- ***Figure 4.2: Production Operations Overview***

  Shows factory communications, test/inspection, in-process control, in-process monitoring, facility control. Shows future capabilities of CAD/CAM on a broad scale stressing the integration of CAM to all aspects of the Design/Production process. Shown also is a schematic of shipyard facilities in order that consideration of CAD/CAM functions described can be related to shipbuilding production sectors.



Shows factory communications, test/inspection, in-process control, in-process monitoring, facility control. *Shows future capabilities on a broad scale*

FIGURE 4.2 PRODUCTION OPERATIONS OVERVIEW

0 *Figure 4.2-1: Integrated CAD/CAM Component Subsystems*

Stresses data flow between identified major factory areas: planning/management, product/process engineering, parts manufacturing/test, final assembly/test, storage/handling. Shown are the many data bases and intercommunication via internal networking services.

o *Figure 4.2-2: Flexible Manufacturing System - FMS*

The use of computers to interrelate several machines on the shop floor to expeditiously effect complex operations on small batches of machine/assembly jobs. The automatic machine tools and/or Fabrication Equipment are linked together by an automatic material transport/handling system, which enables rapid completion of parts, savings in number of machine tools over standard shop floor arrangements and conservation of floor equipment.

0 *Figure 4.2-3: Concept of Hierarchy for CAD/CAM Computers*

Effecting all of the scenarios is the way computer architecture is configured to enable expedient direction, via computers and software, of all operations. Programmable controllers CNC devices on the shop floor and communication links are shown together with their relationship to larger "Master Control" computers.

o *Figure 4.2-4: Integrated CAD/CAM Command Nomenclature*

Relationship of the manufacturing facility, shop area, machine cell, work station, and equipment are shown. Communication nodes and networking and detaiIs of computer use on shop floors are aIso shown.

o *Figure 4.2-5: Distributed Numerical Control - DNC*

Effective use of communications to link network elements of a CAD/CAM system via the levels of computer control outlined in other scenarios is shown. Ancillary features of data feedback from the shop floor for MIS, Inventory Control and Maintenance Functions are highlighted.

INTEGRATED CAD/CAM COMPONENT SUBSYSTEMS

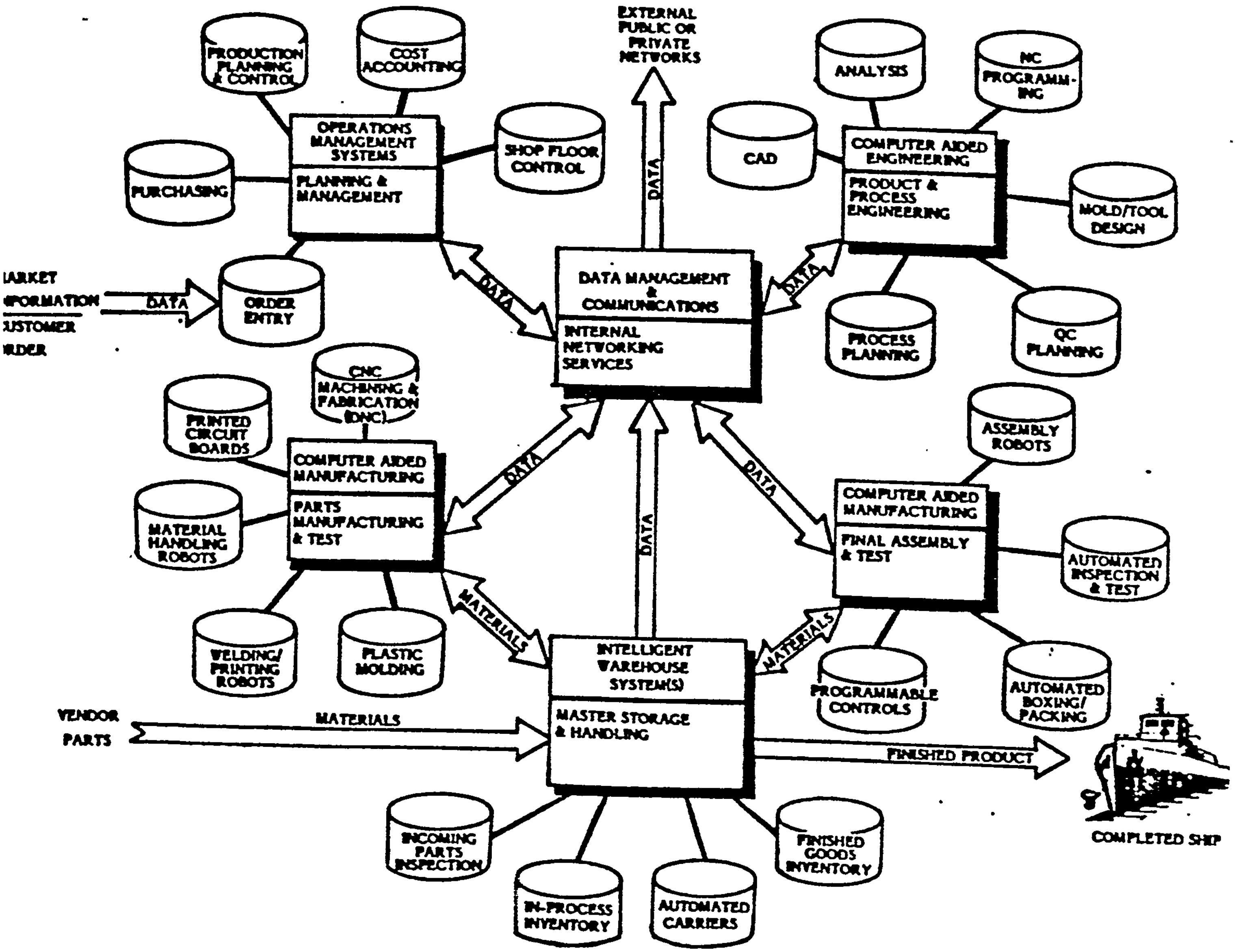
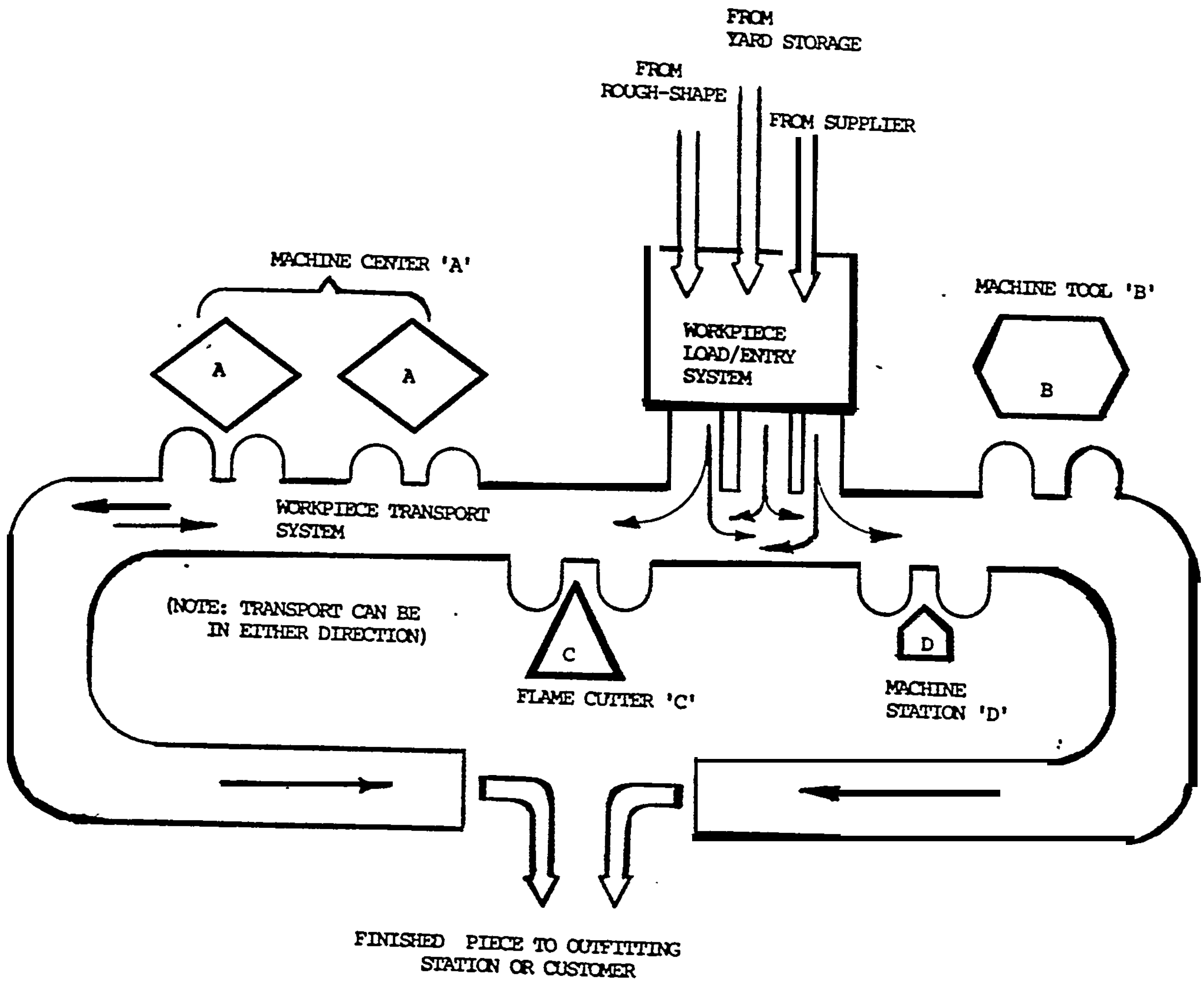


Figure 4.2-1: Integrated CAD/CAM Component Subsystems

Stresses data flow between identified major factory areas: planning/management, product/process engineering, parts manufacturing/test, final assembly/test, storage/handling. Shown are the many data bases and intercommunication via internal networking services



The use of computers to interrelate several machines on the shop floor to expeditiously effect complex operations on small batches of machine/assembly jobs.

Figure 4.2-2: Flexible Manufacturing System - FMS

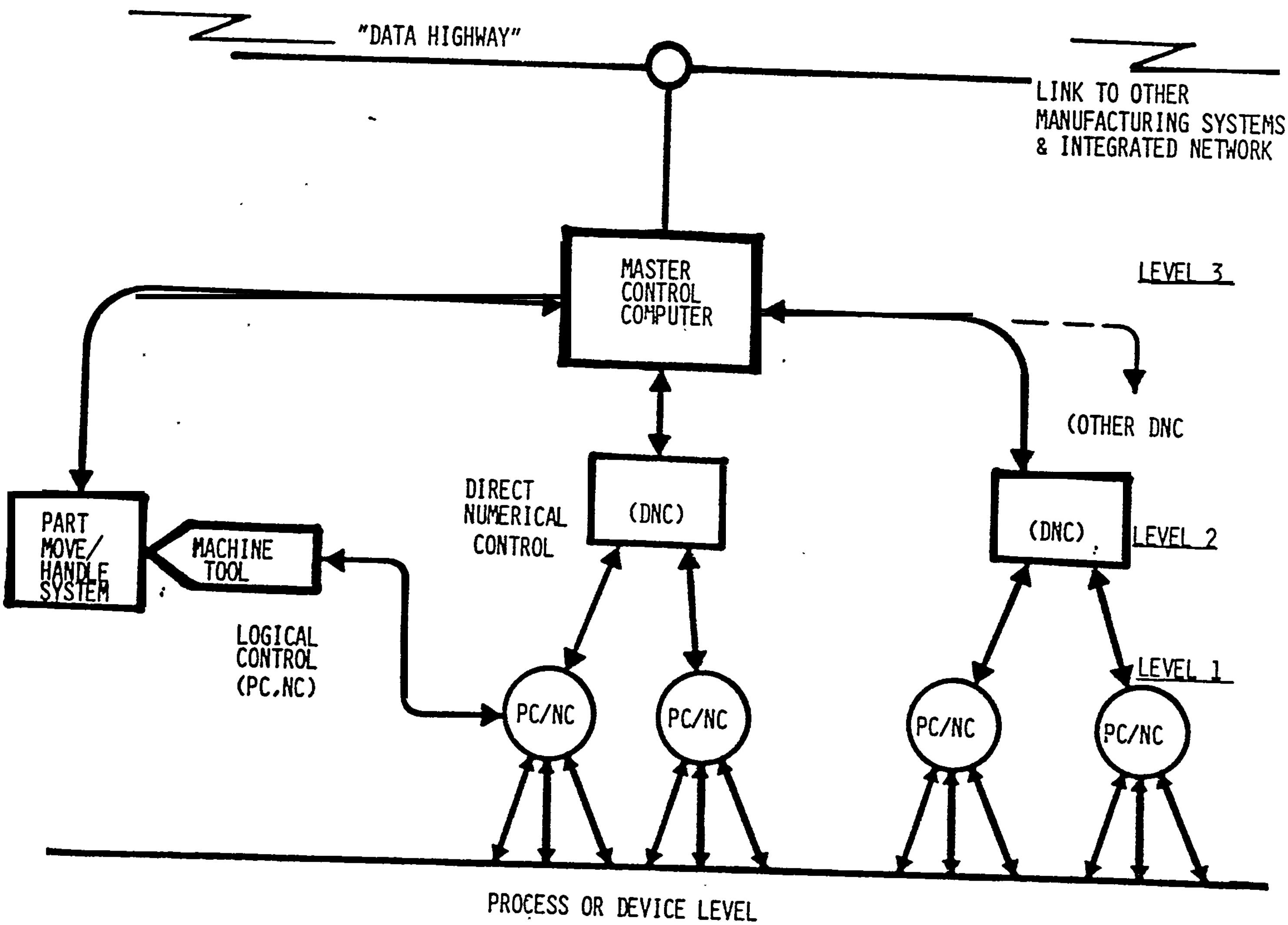
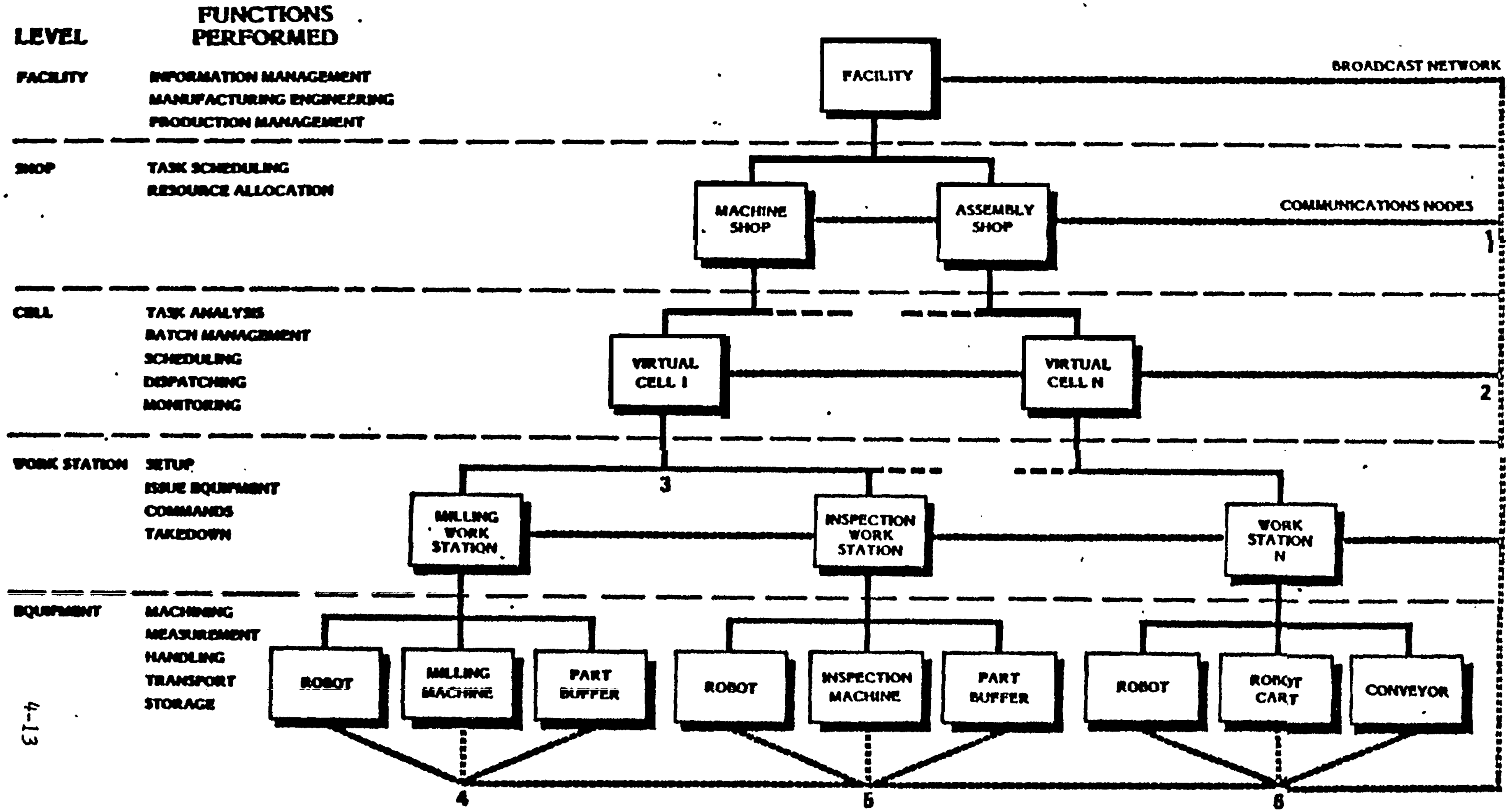


Figure 4.2-3: Concept of Hierarchy for CAD/CAM Computers



The National Bureau of Standards' Automated Manufacturing Research Facility has five level command: facility, shop, cell, work station, and equipment. The control hierarchy is depicted in boxes with solid lines showing the flow of activity from the facility level at the top to specific pieces of equipment at the bottom. Each function box, be it a machine shop, milling work station, or robot, has its own set of controllers for its internal control processes. All the function boxes communicate along a facility broadcast system (track lines).

Figure 4.2-4: Integrated CAD/CAM Command Nomenclature

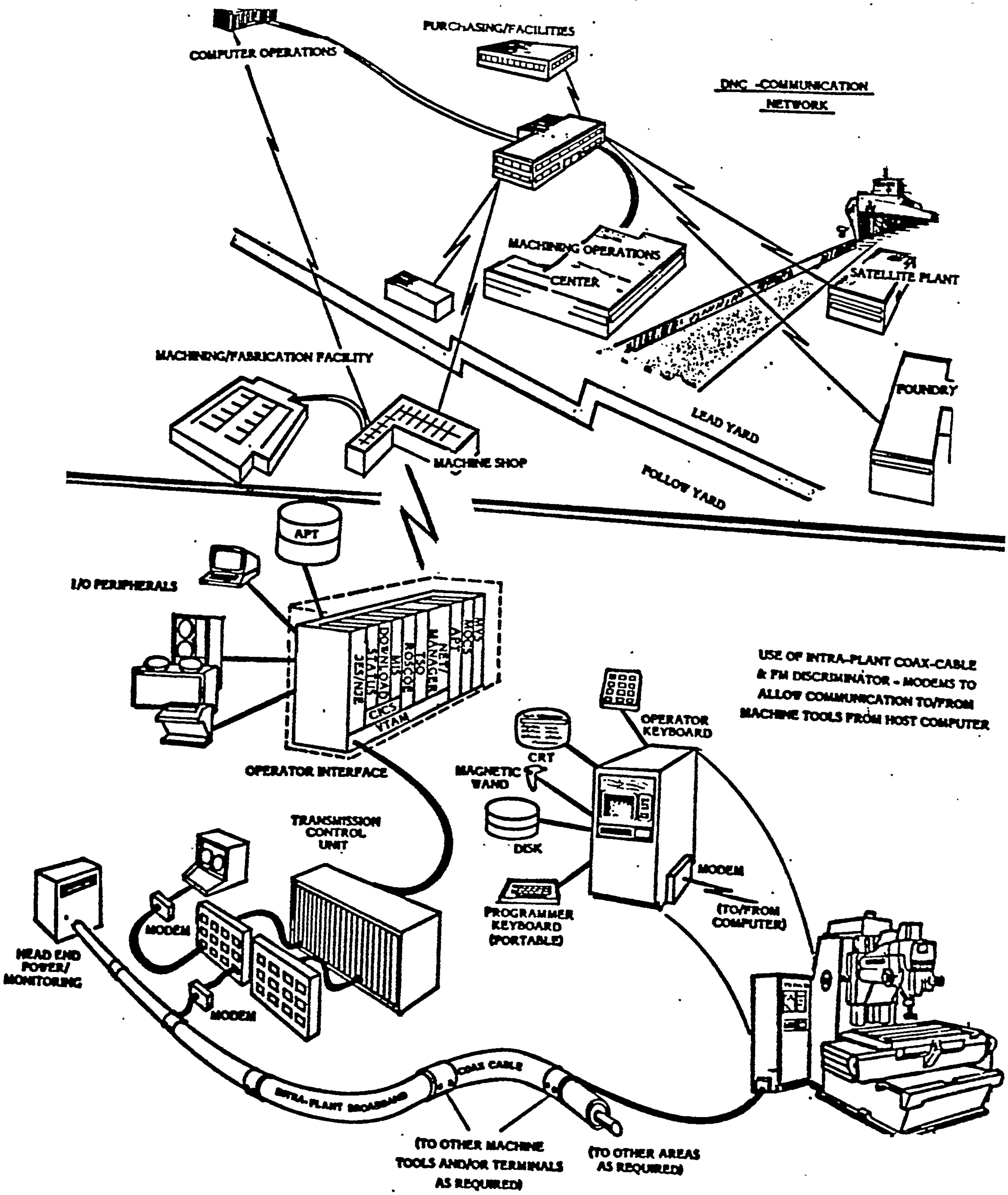


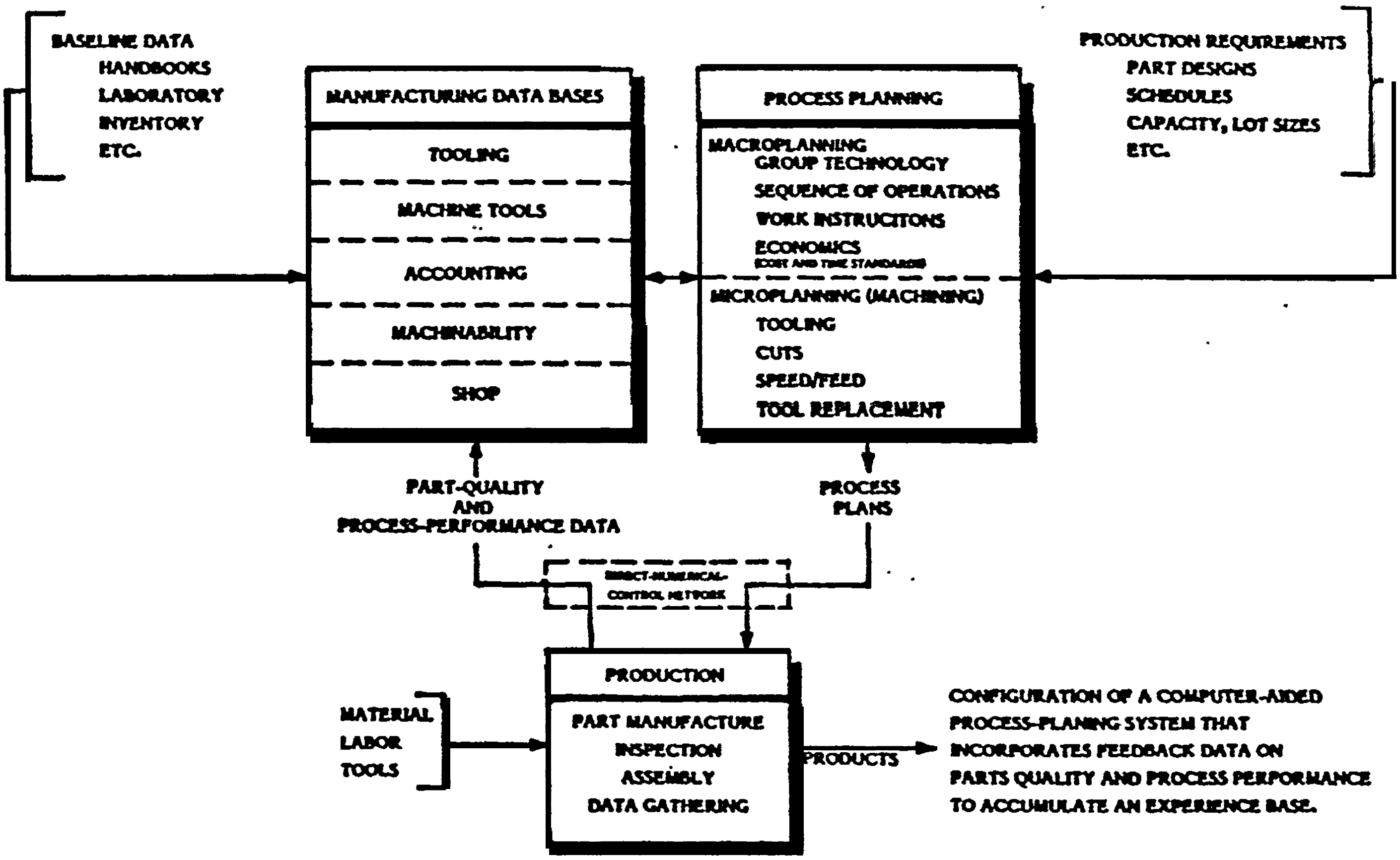
FIGURE 4.2-5: Distributed Numerical Control

- *Figure 4.2-6: Integrated CAD/CAM Process Planning*

  A methodology of using DNC as a means to promulgate quality and performance data, which is gathered while doing actual work tasks, to permit loading of manufacturing data bases automatically. This enables realistic, cost-effective process planning systems to be developed, thus incorporating process planning into CAD/CAM Integration.

- *Figure 4.2-7: Activity Structure for New Generation Shipbuilding Practices*

  A visual chart showing interrelationships of zone outfit planning (ZOPM) terminology. Basically this figure is a classification/outline of several ZOPM manuals which served as a guide in selecting/analyzing other scenarios in order that a shipbuilding orientation be maintained.



Methodology of using **DNC as a means to** *promulgate quality and performance data gathered while doing tasks to effect loading of manufacturing data bases enabling cost-effective process planning..*

FIGURE 4.2-6: Integrated CAD/CAM Process Planning

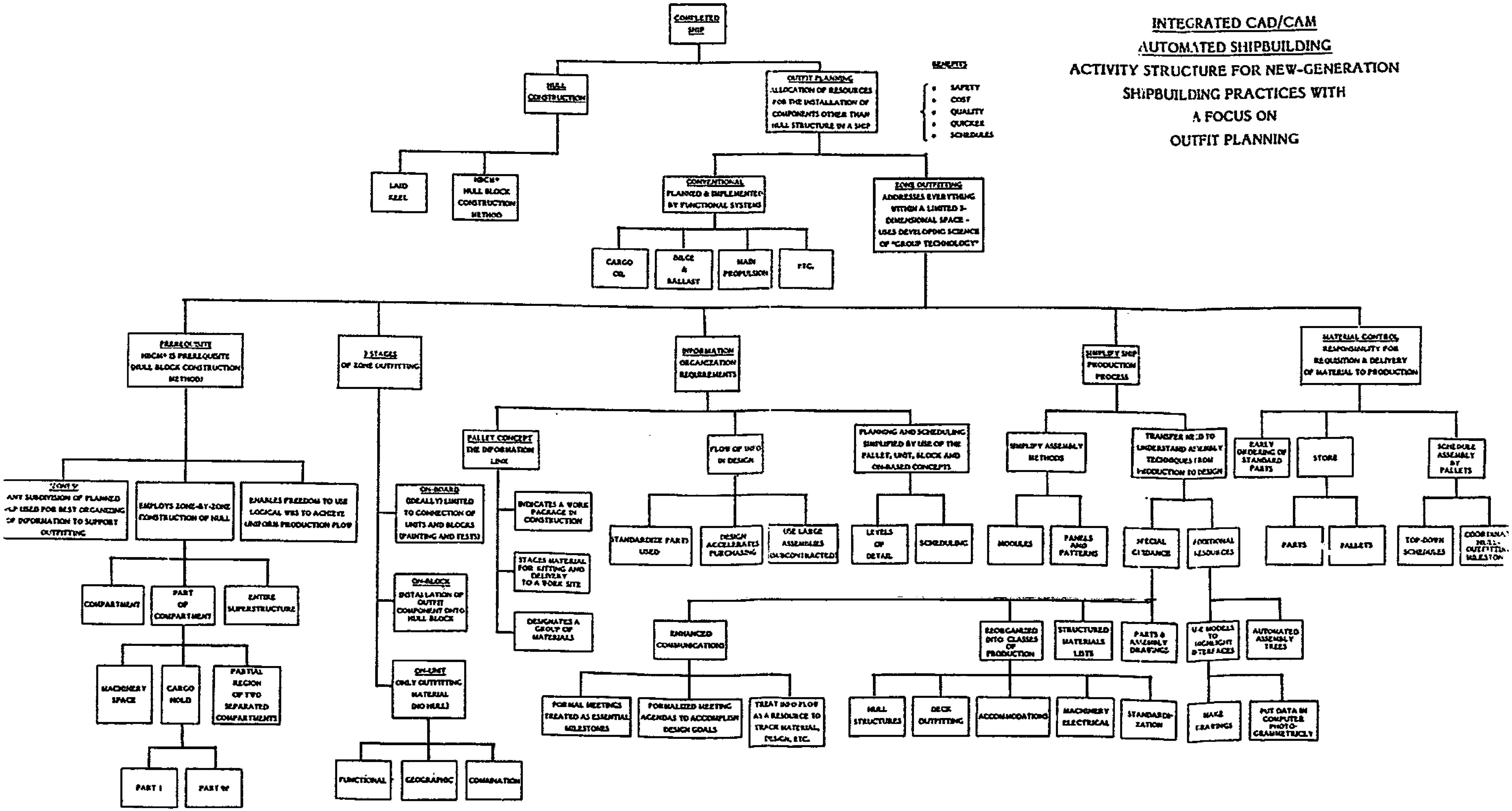


FIGURE 4.2-7: Activity Structure for New Generation Shipbuilding Practices

4.3 UTILIZATION OF SCENARIOS

Scenarios were used to effect a transfer and integration of technology, using the process depicted in Figure 4.1-3. During on-site visits they served as narrative simulation aids, with comments and changes being noted directly on scenario sheets to aid in formulation of forecasts of shipyard CAD/CAM usage. In many instances, the actual viewing of a scenario was unnecessary, as it. was found to be non-applicable or that substantial shipyard available data was used to outline projections. In the latter case, data noted from discussions was analyzed and elements of information noted on the most nearly appropriate scenarios. As explained earlier, by noting features on pre-formatted scenarios, the proprietary nature of specific hardware /software use is protected.

Narrative simulations were carried out at the end of interview sessions by using scenarios to prompt questions. Shipyard comments depicting expected future uses of CAD/CAM were noted on appropriate scenarios, and in turn promoted questions. These questions stressed the simulation of yard operation under the newly restruct-ured- scenario conditions, with the "What If" queries largely coming from the graphic interrelationships within each scenario and the relationship between the selected scenarios. This mode of fleshing out the scenarios provided a rapid means of expanding CAD/CAM integration forecasts after the initial question and answer

period. At this time, the focus on computers and software became very evident and questions on how software would be acquired (make-buy-lease) were introdu-ced. Software engineering issues were also focused on at this time. These issues are presented in the next section.

5. SOFTWARE REQUIREMENTS FOR DATA DRIVEN AUTOMATION

Design/production integration has as its objective the provision of an integrated structure to enable full exploitation of modern computer resources in shipyards, ship design agencies, interfacing governmental groups and supporting subcontractors. A critical sub-goal is to reduce the time for acquiring these systems, reduce their costs and increase resultant benefits. Basic to this task is system definition and system integration. Since the systems task is centered on required application of computers, the need for large-scale special purpose software, and applicable integration techniques is necessary. Effective system definition, creation, and integration to fulfill these needs depends on utilization of modern software engineering techniques, methods, and tools.

Thus, the single most critical need of the CAD/CA.M integration process is the application of software engineering, with its associated tools and methods, to the integration process over time. The development of a shipyard software engineering capability, and the accompanying software tools and the knowledge to use them is critical to CAD/CAM integration. Since many suppliers of CAD/CAM equipments do not employ highly trained specialists, some industry experts predict that most of the software will actually have to be rewritten. .

5.1 SOFTWARE SYSTEMS AND INTEGRATION

Many of the problems associated with integrating CAD/CAM systems are being solved. The eventual introduction of software systems and system integration techniques to shipyards on a large scale will have a substantial impact on both the quality and scheduling of these efforts.

Impetus for the integration of CAD/CAM systems has expanded in virtually all industrial areas in the past decade. Much of this expansion was trigged by the basic realization that the relationships of the CAD and CAM systems required centered on their software component, and the need to manage the subsystems of each as attributes of computer-based systems. The combined efforts of CAD/CAM hardware vendors and technological advances in computer systems have already introduced software as a central theme in CAD/CAM integration, but there is still *a need for understanding the software engineering process on the user level. The true value of integrated CAD/CAM systems can only be realized, maintained and improved by further development of improved shipyard user-defined software systems and effective employment of system integration techniques. These basics are the backbone of a successful in-house, practical level,' CAD/CAM integration program. The development of a high-yielding, improved productivity, CAD/CAM system depends on successfully grasping the software engineering and management techniques needed to integrate the emerging computer-based systems. Basic to an understanding of software engineering, and the use of software tools, is an understanding of the attributes of computer-based systems.*

5.1.1 *Attributes of Computer-Based Systems*

A computer-based system is any arrangement of devices that can produce a change in a defined environment through direct, or indirect, control by a digital computer acting under the guidance of data interpreted by a set of software produced instructions. Computer-based systems can impact the physical environment of the shipyard in many ways, as exemplified by automated machines such as large-scale numerical control flame cutters. Until rather recently, the behavior of computer-based systems was studied, and viewed, in isolation from outside areas. Notably, the CAD, CAM, and MIS functions all had particular "subsystems" with stated functions, limited users, and known outputs. However, computer-based systems are now being used to provide dramatic changes in the CAD/CAM areas of industry, and the newly affected domains of MIS functions, standardization practices, and productivity methods. Much of this change has risen from recent improvements in the techniques of defining and integrating computer systems using both new approaches to define software, and new tools to manage software development and use. A few key concepts of computer-based systems can make valuable contributions to CAD/CAM integration and to the understanding of the integration process. Moreover, the direct transfer of these concepts can apply the technology of software and system control at a level where managerial decisions can be made to control quite precisely the conditions, costs, and construction of these systems in the shipyard.

Computer-based systems, and this includes CAD and CAM systems, all share common attributes. The importance of these shared attributes in the CAD/CAM integration function is that methods of identifying, isolating, and dealing with problems in other industrial computer applications can be transferred to the shipyard CAD/CA,M integration process. These attributes of computer-based systems are:

- *Computer-based systems cannot be viewed in parts. There is no "hardware part" and "software part" of a system. This mistaken view, commonly taken, leads to confusion and inappropriate definition of systems. An accurate representation of systems currently in-place, in development, or projected for the future requires a total system's vantage point. Dealing with the CAD/CAiM integration process requires techniques and tools to provision this total system view, specifically. adapted to the shipyard needs in the production management, and engineering environment.*

- *Computer-based systems, as planned or as existing, outwardly have substantive differences in appearance, function, and operational role. These systems may be* ***CAD, CAM, MIS, Management Reporting,*** *accounting or subsystems of larger CAD/CAM systems. Whatsoever their description, the in-house effort required to purchase, lease, design, and/or create a computer system must be viewed and treated with the same management perspective. Underscoring this point is the requirement to cope with the synergistic effect of CAD/CAM integration, wherein the software integration process affects many areas outside of the usual CAD/CAM arena. This attribute of computer systems, making it possible to deal with all computer systems through one management approach, while accommodating the operational characteristics of a specific shipyard, gives a cost effective basis to cope with the new interrelationships.*

- *Computer-based systems all demand careful attention to detail during their development process. In particular, the development of software requires a development approch as methodological and concisely* *defined development approach as does a complex, new hardware project. Shipyards can benefit from instituting and using rigorous steps to develop software in all areas of CAD/CAM integration, as well as ancillary computer systems activities.*

- *Computer-based systems are largely composed of software, which has as a chief attribute the element of intangibility. Software cannot be seen, touched, or physically recorded by human-understandable optical means. As a result, the only way to transmit information about the software content of a computer-based system is via documentation describing the system and its attributes. Use of a comprehensive system and software documentation methodology is a key to economical development, use and enhancement of software and systems in shipyards.*

Software is clearly a very critical element in all computer-based systems. Yet, many shipyards have not made significant efforts towards coping with even the rudiments of software development disciplines. The challenge of CAD/CAM integration, due to its innate complexity, has given new impetus to this task. An added benefit of undertaking formal institution of a software engineering effort within a shipyard environment will be a better, more cost effective, mode of coping with all required software, due to the attribute similarities of computer-based systems.

5.1.2 Software Requirements Conception: The Starting Point

Critical to an understanding of how to deal with software more productively, is the ability to conceive of how the need for a particular software system begins. A practical schema to place this in the context of CAD/CAM integration must consider the system as a whole, and the role software vis a vis hardware plays in the systems conception and development process. Software requirements for CAD/CA,M integration emanate from the CAD/CAM system requirements. Importantly, these same system requirements are the starting point for hardware development. The relationship of these two requirement activities is seen in Figure 5.1.2.

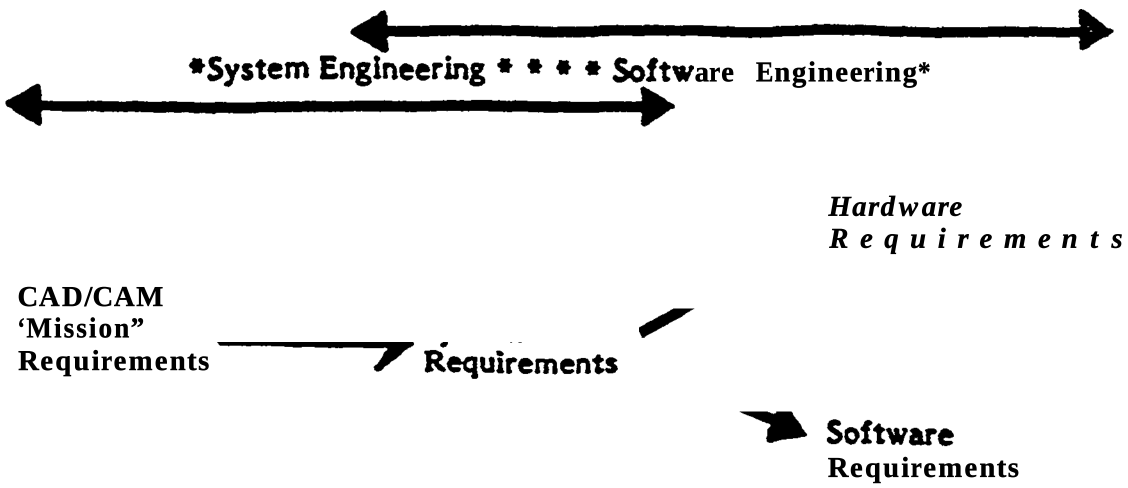


Figure 5.1.2 Evolution of Software Requirements

As depicted, the investigations of alternate approaches to resolution of CAD/CAM requirements is analyzed and designed by the systems engineering process. System requirements are broken-down into the hardware and software requirements process. The interface of system engineering and software engineering is the overlap resulting in the software requirements.

Clearly, to optimize development of CAD/CAM systems where software is an end product, the need for application of software skills starts with the systems requirements. This applies to systems which are purchased, leased, or built in-house, since all of these systems include potential trade-off decisions which can affect an overall integration plan.

The differences between Hardware (HW) and Software (SW) are important to both development and final end product of the systems development process. These differences are highlighted by the following comparisons:

- *HW can deveIop prototypes (Breadboards) for requirements --SW is created in a single, continuous process.*

- *HW can be easily differentiated and identified — SW is intangible and cannot be identified.*
- *HW depends on using standard components, often of pre-proven quality and operation -- SW depends almost entirely on a "First-Time" approach for every effort.*
- *H W manufacture is a production-oriented function -- SW is largely characterized by activities more akin to research and development.*
- *HW quality standards are universally understood and easily verified -- SW has no set quality standards.*
- *HW design is finalized before it is built -- SW design is in a constant state of flux.*
- *HW design/construction, if poor, is visible by simple inspection -- SW, if poorly designed, is often not detectable as poor quality (until too late).*
- *H-W development is understood by management personnel -- SW development generally is not.*
- *HW changes are not easily affected - SW changes, for better or worse, are quite easy to introduce.*

This comparison of HW and SW points is outlined to emphasize the many clifferences between HW and SW. However, both emanate from the system requirements, and the relative stress placed on HW and/or SW both start at the same time as the systems requirements process. Thus, early start of the SW process is critical to viable system design and cost effective system development.

5.1.3 Shipyard CAD/CAM Integration; A System Focus

Important to the conception of the design/production integration process for shipbuilding is the understanding of the scope of the systems effort considered. Islands of automation can be considered subsystems in the CAD/CAM integration picture, but how the system as a whole fits together and interacts the totality of these subsystems, is the approach on which to base an integrated concept. Each of the scenarios considered in Section 4 can be considered a top-level systems approach to the CAD/CAM integration process. The eventual systems approach taken can be a variant of any one of these, or a completely different approach. The point to be underscored is that the systems engineering process is essential to defining comprehensive systems requirements, which should reflect, regardless of how broadly, a total approach to the CAD/CALM integration process. From this vantage point the system requirements, the hardware/software requirements, can be related to aspects of the overall system. Critical to this approach is the ability to trace changes in technology at the systems level to required changes in software requirements. Likewise, a deficiency in software development must be traceable to current, or future, system level components.

The study of software productivity for CAD/CAM integration, through the use of scenarios, adheres to this systems engineering philosophy. This rationale gives both a rigid framework of logically introducing new technologies, and an opportunity for individual uniqueness in CAD/CAM integration. This is in keeping with the nature of the systems approach.

5.2 SOFTWARE DEVELOPMENT & MANAGEMENT TRENDS: HISTORICAL PERSPECTIVE

Among the most challenging problems in CAD/CAM integration is the detailed development of the functioning software that will actually bind the disparate sectors of the integrated CAD/CAM process together. It has been recognized for several years that the software aspect of the integration process is a pivotal element. In the absense of an encompassing software capability, the myriad functioning of individual "islands of automation" will not interact with each other to sustain a meaningful level of productivity. In the past decade it has also become clear that software methods and tools can act as catalysts in the software development process, accelerating developmental steps and improving both software productivity and quality. A detailed knowledge of the role these tools play in software development is an essential precondition to their use. Additionally, a synopsis of why they have become important to management of the system development process is critical to grasping their importance vis a vis the ever changing world of software and computer technology. A brief historical perspective of the growth of software engineering and associated software tools usage as a response to the growing computer architectural and software sophistication will serve this need.

5.2.1 Software Accomplishments and Challenges

In order to appreciate the power of software engineering methods and tools and their role in the shipbuilding CAD/CAM process, it is helpfuI to understand the historical changes in focus of software development in response to changes in computer technology. Succinctly stated this computer history can be viewed from afar as a legacy and a challenge. The legacy of the 1970's was the ability to engage very large software systems on large, central computers. The challenge of the 1980's has become the matching of user oriented software to proper computer hardware. Distributed processing, made possible by mini and micro computer technology, has become "the means to this end. However, recognition of the need to treat these growing systems capabilities as functioning, integrated systems, has given a new perspective to these developing areas of technology. This view holds that distributed processing be recognized as a subset of centralized processing *because the larger problem, of which a distributed system is a part, is the system as a whole which must be studied and suitably partitioned. This is the systems approach to CAD/CA.M integration for shipbuilding recommended in this study.*

Responses to these computer technology trends, to enable effective software development, has evolved in several stages. Essentially, these stages are characterized by the growing recognition that the mystique of programming must be redirected to sound management practices based on a comprehensive systems approach. The 1970's reacted to this need by emphasizing software documentation (program to program); 1975 saw the emergence of software engineering (program to user). The challenge of the 1980's has emerged as more software engineering, with a focus on developing better software requirements and software specifications.

To accomplish this end, software tools and techniques, as well as management techniques, are needed. A new skill-type has emerged to provide these needs, the Software Engineer. Key to the CAD/CAM integration process are the skills provided by software engineering.

5.2.2 *Software Engineering*

Software engineering is defined as the science of design, development, implementation, test, evaluation, and maintenance of computer software over its life cycle. Thus the software engineering process is aimed at designing software systems to make them more producible. A software engineer strives to make software design development, test and maintenance less labor intensive through the use of software tools, software management systems, advanced programming methods, communication interfaces and automated analytical aids. These are applied by the software engineer in a disciplined order through skillful use of suitable tools and methods to create practical solutions to a user's documented problem.

A software engineer's focus is the entire life-cycle of a software effort, which starts from the system definition phase through the maintenance and update of computer programs. A software engineer is involved at the very earliest phases of *a task, and while his presence is not required after system completion, his legacy should last forever. This legacy is the remaining management system comprised of software tools that enables continual, high-quality, monitoring and enhancement of software throughout the system's usefuI life cycle. The specifying and development of these software tool based systems is the principle activity of software engineering.*

An overview of Software Engineering history is presented in Figure 5.2.2.

How the Software Engineer works in the context of an existing organization, and who he interfaces with determines the extent to which software tools can benefit management. A description of this role follows.

5.3 SOFTWARE DEFINED AND UNDERLYING ISSUES

Integrated CAD/CAM systems are becoming increasingly software dependent. Not only are more applications using computers, but the complexity of computer programs is expanding radically. Unfortunately, this software explosion in CAD/CAM integration has created development and reliability problems that are getting beyond control in even the best prepared industries. A basic knowledge of fundamental software definitions and underlying issues will underscore the importance of using software tools in future shipyard efforts.

5.3.1 Software Defined

Software is defined to be the totality of instructions, or software package, required to create an intended function with a digital computer. Such a software package is called a computer program.

TABLE 5.2.2 History of Software Engineering

Late 1960's:	*Software development process runs out of control, description of "Software Engineer" coined during a NATO meeting of 1968-1969.*
1969-1971:	*Software engineering principles developed, and focus on listing of good programming practices. Rigorous definition of the programming development process begins.*
1972-1973:	*Structured programming and development of desirable programming styles. Focus on delineation of where errors are made in programming.*
1974-1975:	*Exhaustive testing and risk areas investigated. Software tools to automate software testing emerge.*
1976-1977:	*System requirements linked to software requirements, methodologies for design and specification aids. Software cost modeling emerges.*
1978-1980:	*General acceptance and practice of software engineering, especially growth of software tools use. Expansion into new computer hardware architecture areas of net working, mini/micro system and distributed processing.*
1980-1982:	*Increasing use of tools to aid long-term control of large software systems through software management systems. Development of software support centers to automate maintenance of software.*
1982-1984:	*Study and application of software engineering to life-cycle management of large, automated systems in industry to develop software tools that reduce the total cost of ownership of large, computer driven systems.*

Computer programs are manifest only by the physical media on which they are stored. This can be punched cards, mylar or paper tape, a magnetically encoded tape or disk pack. However, the actual creation of a computer program, a software package that performs a specific function on a designated, or target, computer, is not directly observable in the real world. This is so because the particular series of encoded software instructions varies depending on the peculiarities of the computer system on which it is designed to run, or operate. Accepted methods of encoding are termed software languages. The software language used for a computer program is limited by the repertoire the target computer hardware is designed to accept. Software languages can be converted, or changed in structure, to other forms of computer language in some cases. The original function of encoding software to create an application program is termed programming; changing of one software language to another is called transition, automatic changing of one software format to another, using a tool called a translator, is called translation. These processes, for the purpose of this report, will constitute the definition of software, and software programming.

Software documentation represents a means of describing all stages of the software development process to a wide range of users of software programs, as well as participants in the development, test and operational phases. Documentation includes user manuals, program operation instructions, program listings, schematics, flow charts, management descriptions, system specifications and all other descriptive printed data about the program, its' development, testing and operation.

Software has, by way of software engineering, evolved a generic term which describes the smallest defined portion of a program which has interfaces. This is the computer program module. A module represents the smallest definable unit of work for a programmer, and as such is the basis for programming tasks, management of change and testing. A module has size, expressible in Lines of Code (LOC), and a relationship to other higher-level (parent) modules and lower-level (child) modules. Modules can be given names, usually a reference code keyed to their relationship in the hierarchy of a program, so they can be managed and controlled as mini-packages of software. Software engineering and software tools *start controlling software at the module level. Software modules can be assigned schedule dates for coding, de-bug, integrated test, etc. They can also be assigned resources, such as a responsible analyst, programmer, etc.*

The Hierarchical Item Description, or 'HID' designates by use of a reference code, a unique software module. Typical module descriptors and the relationship of a module to a software "tree-structure" or hierarchy are shown in Figure 5.3.1. Software engineering deals with means of facilitating, the design, creation test and use *of software modules.*

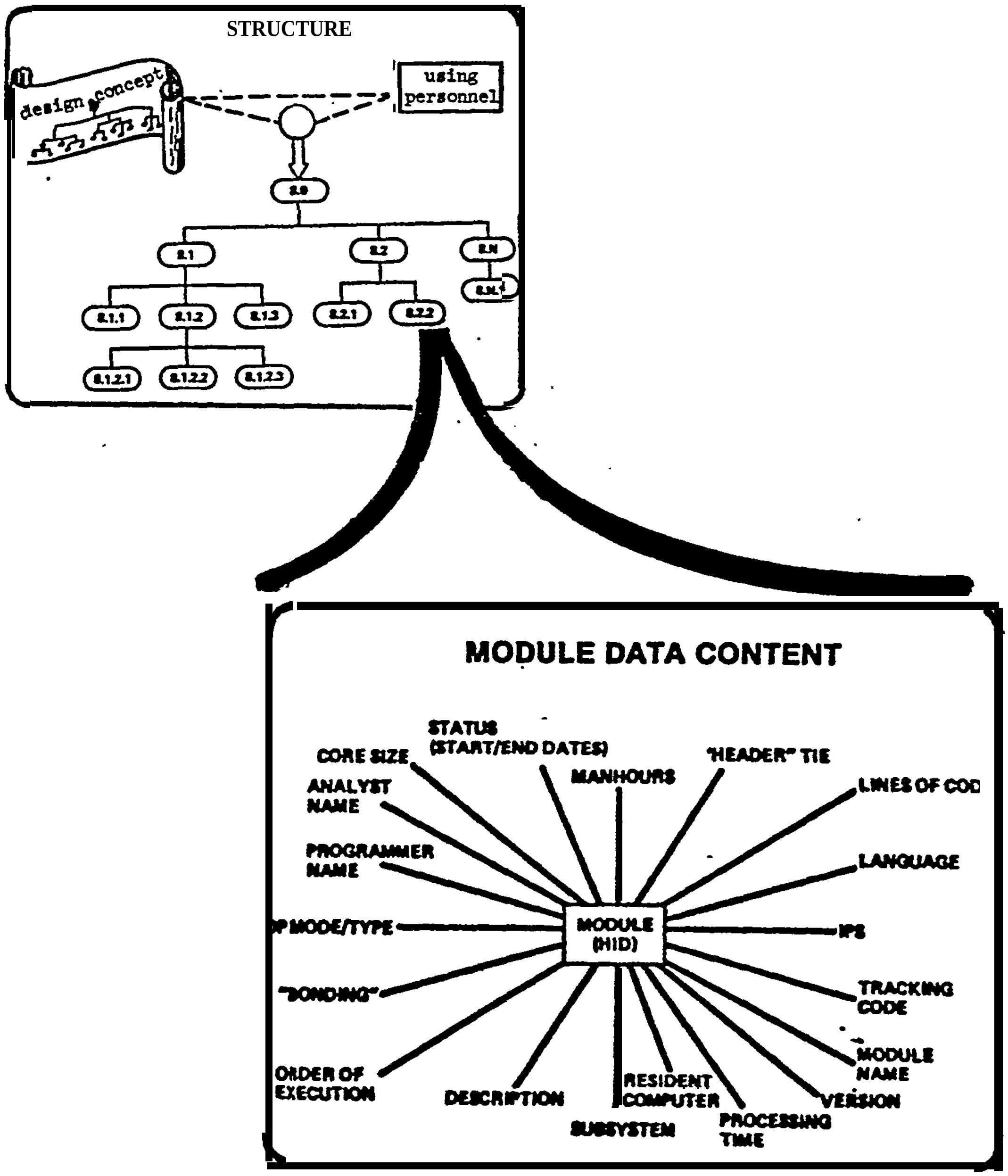


Figure 5.3.1: Module with examples of assigned resources.

5.3.2 *Underlying Software Issues*

Both the use and understanding of software tools as an important adjunct to modern software development have more to do with unseen, or at least understated features of software than with obvious software functions. Underlying issues of software are those features which effect the complexity, use, cost and longevity of a software application program, but are not necessarily a part of the application software package itself. The CAD/CAM integration effort in shipbuilding must recognize and cope with these underlying software issues to effect a viable design/production integration process. Software tooIs deal with these underlying issues as much as with the visible software effort and finished programs themselves. This aspect of software, where much supporting effort is required for the smaller visible software effort, is oftern referred to as the "Software Iceberg".

The Software Iceberg is a very real model of in-house, leased or purchased software. Any decision on a software acquisition project must seek out and examine the Iceberg phenomena to fully grasp the real costs and technical issues of a software package. Figure 5.3.2 is a schematic of this "Software Iceberg" concept.

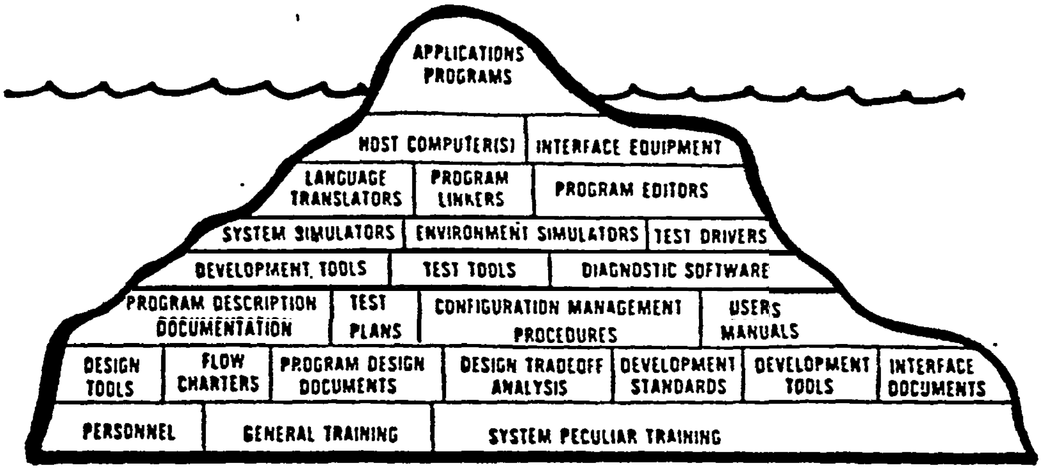


Figure 5.3.2 : The *"SOFTWARE ICEBERG"*

Depicted are the unseen, yet required, features of software which need attention and resources for a system to work.

Depicted are the unseen, yet required, features of software which need attention and resources for a system to work.

Costs of software are another trend which have made it critical to examine both the efficacy and productivity of software in the CAD/CAM integration process. Figure 5.3.2 .-A shows the software cost trends compared to computer hardware costs.

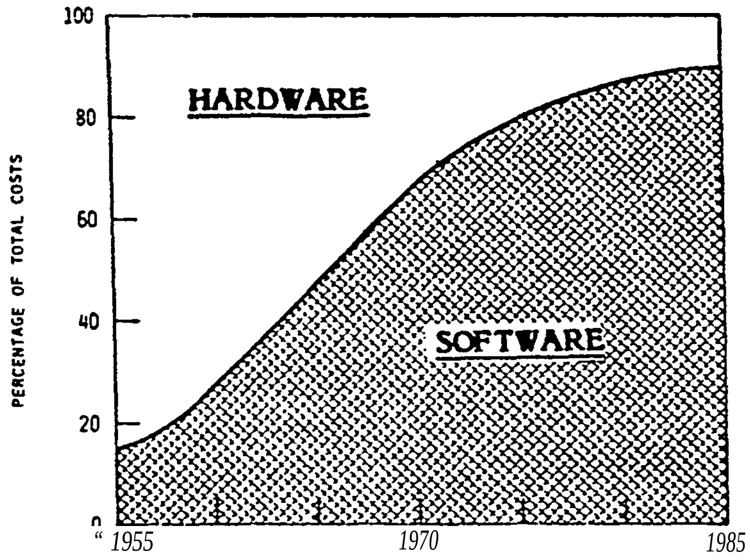


Figure ***5.3.2-A*** ***SOFTWARE AND HARDWARE COST TRENDS***

Shown is the trend of software vs. hardware costs. Increasing costs of software are obvious, with the increase due to higher costs for both new software and increasing costs for software maintenance.

The underlying issue here is clear. Lower costs of hardware are making it more cost effective in terms of number of bits of processing capabilities acquired per dollar expended. However, this same trend has the result of requiring more software. This trend of cheaper and more powerful hardware has another, more insidious effect on software. Hardware systems are more specialized and increasingly "mini/micro" computer based in a stand-alone mode, and the tendency to update computers becomes more tempting. All of these trends require software, and unless existing software can be translated or converted, many new application programs will have to be written for the newly acquired hardware.

As computer based sub-systems are added to a larger integrated system, the capacity of the central computer coordinating such a network becomes overtaxed. In an effort to conserve on hardware updating, often an attempt is made to maximize utilization of existing computer hardware by utilizing as much of the available capacity as possible. However, as the capacity of computer hardware is approached, the productivity of software, and thus the cost, is also adversely affected.

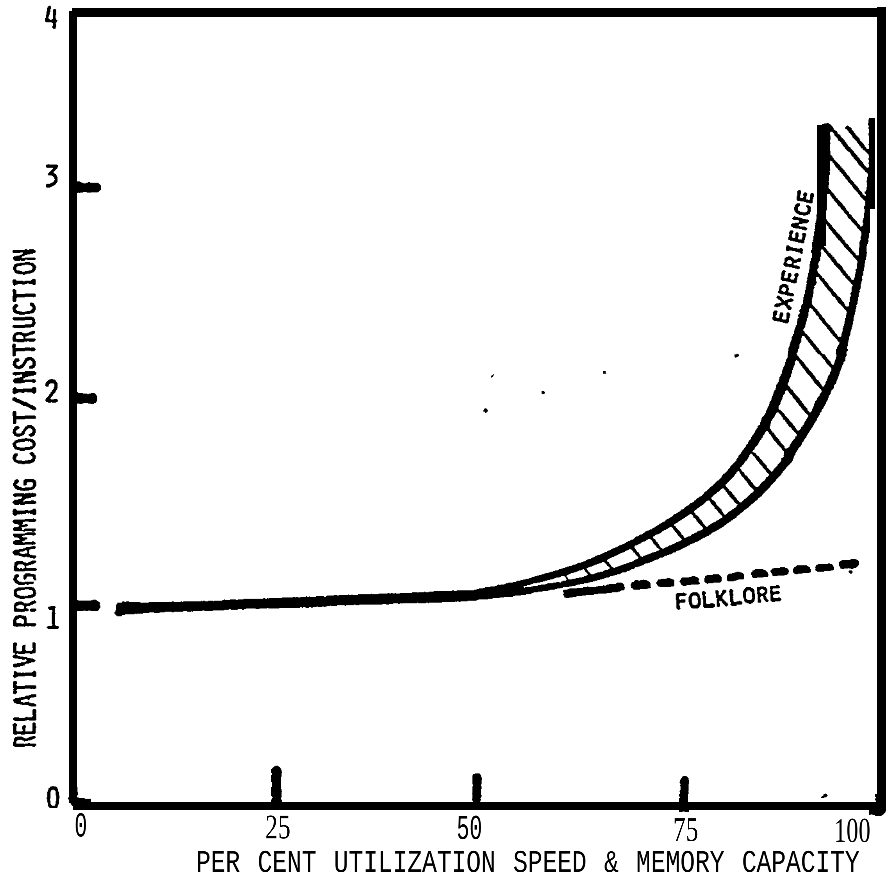


Figure 5.3.2-B: ***HARDWARE CAPACITY EFFECT ON SOFTWARE***

Figure 5.3.2-B clearly shows that there are many underlying decisions to be made as a system requiring software grows. If the same hardware system is maintained, the cost of adding additional increments of software grows as hardware capacity is approached. Additionally, system capability for response time degrades as the system size gets bigger. If hardware is updated, translation/transition of software, at no small cost, is required and resources must be expanded.

These underlying software issues are all indicative of less apparent, but still very real negative effects of the previously explained growing technical improvements of computer hardware, and its production at increasingly lower prices. This improvement in hardware has been achieved through automation of the production cycle of hardware devices. However, software, not unlike shipbuilding, is extremely labor-intensive. This labor intensive quality applies to both the maintenance and enhancement of existing software, as well as the creation of new software. Large organizations expend over one half of their allocated software resources on maintaining/correcting existing software. This is a characteristic of American Industry as a whole, where a quarter of a century of largely antiquated software is being patched together by a cadre of programmers. On the national scene, this has become an underlying software issue. International software markets are being lost to other countries because new entrants are gaining on the United States in the implementation of new software through employment of totally new software development methodologies.

This issue of the United States losing its productivity edge in new software technology could initiate many deleterious side effects. The use of automated tools to affect viable integrated CAD/CAM systems in any industry by other countries will give them a state-of-the-art technological lead over the United States. Thus, the continued dependence in this country on older software methodologies in the face of emerging computer hardware technologies and the need for CAD/CAM integration could create a situation where the United States would loose ground in shipbuilding technology even while introducing modern machines and facilities. A focus on automation of software development through applied software engineering and software tools is proposed to significantly reduce the risk of this hypothesized scenario actually occurring.

5.4 NEW TECHNOLOGY AND NEW SOFTWARE NEEDS

The development of software engineering techniques for producing software has been important to the actual design and development process. However, the value of software engineering transcends this narrow slice of time in a programs existence, and includes important activities prior to, and long after, initial program build and test. This comprehensive view of software over its entire time continuum is termed the software life-cycle. A grasp of the concept of a software life-cycle approach to systems planning and management .is invaluable to the integrated Design/Production process.

5.4.1 FOCUS SOFTWARE LIFE-CYCLE CONCEPT

A variety of mechanisms have been employed to achieve greater control and standarization of the software development process. However, all too often these specialized methods of coding, or testing, have dealt largely with operational features of a particular language rather than the broader underlying issues. As a result these specialized techniques have not been successful in bringing together the requisite computer, personnel and system skills to affect constructive software systems development.

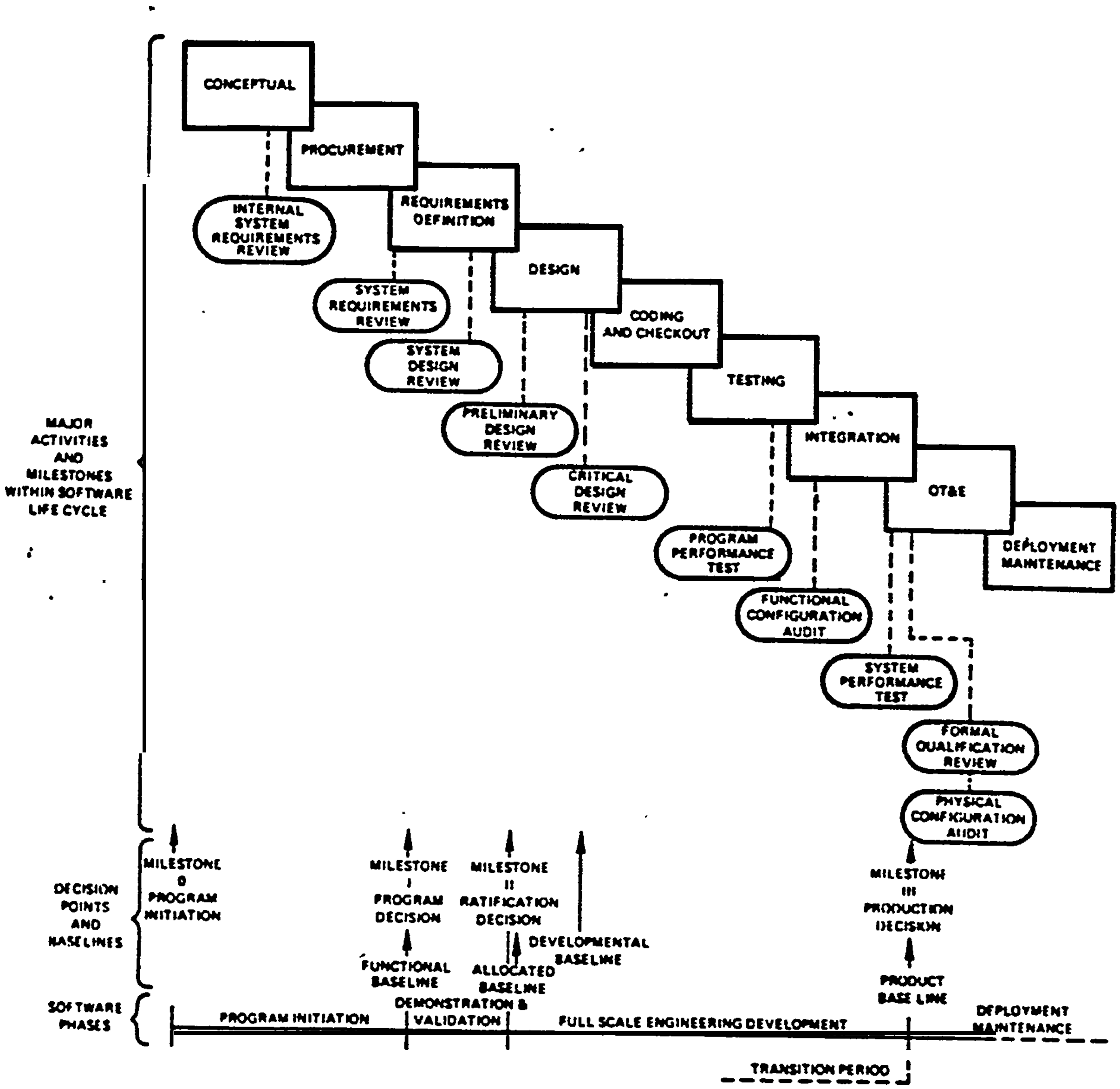
The life cycle approach to software development represents a comprehensive technique of dealing with the many variables of a software project. Emphasis is placed on presenting information on a software project during its entire life-cycle, and dealing with the underlying issues of software, as well as the visible coding and test functions. This is accomplished by dividing the life-cycle into well defined time phases, each time phase with accompanying major activities which are described in detail. This approach enables software management to be applied to a degree that management of in-house programming, monitoring of contract programming and evacuation of purchased/leased software becomes practical. An additional beneficial result of the life-cycle approach is the ease of standardizing software tool use and applications. Most important in this regard is the ability to maintain control of software through the operational phase where tools can aid in facilitating program maintenance and enhancements during system use. This value of software tools, *long after a program has been in use, is a strong feature of the life-cycle approach. The use of such tools, during all phases of the life-cycle, are planned early in a project effort. In fact, the impact of such decisions on a software project are inversely proportional to the life-cycle phase of the project. This means that the most important decisions are made early in the requirements analysis phase. The time phasing and activity descriptions inherent in the software life-cycle approach yield a means to give a project manager enough understanding of the total life-cycle process to enable better performance of detailed planning that will lead to better management of his own staff, and/or software contractors responsible for project development. The result will be improved software cost, scheduling and performance characteristics over the entire life-cycle of a project.*

5.4.2 ALTERNATE LIFE-CYCLE REPRESENTATIONS

There are many variations of the life-cycle concept. All are useful, some more valuable in one setting than others. Determination of the ideal life-cycle for shipyard software development and CAD/CAM integration must be ascertained by considering the site of expected software projects, shipyard capabilities, and management requirements. All these factors are important as a framework for technical, management and budgeting considerations for the software process, and as prerequisites for formal use of software tools. Figure 5.4.2 Depicts Software life-cycle phases, major milestones and the individual life-cycle phases.

FIGURE 5.4.2 SOFTWARE LIFE CYCLE PHASES

INTERRELATIONSHIPS WITH MAJOR ACTIVITIES, MILESTONES AND SOFTWARE PHASES OVER TIME.



This representation of a software life-cycle stresses applications to technical projects. Integration of shipbuilding CAD/CAM will require both highly Technical and MIS/Business computer applications. Whatever the focus, there are many debates over what is the best way to represent a system's Life-Cycle. A basic argument against the Phased Life-Cycle approach, as depicted in Figure 5.4.2, is that it results in a problem of interfacing the software functions developed in each discrete stage. These problems typically do not surface until the end of a project. Figure 5.4.2-A compares a simplified conventional (phased) life-cycle concept with a typical structured approach. Whatever life-cycle approach is adopted must reflect the indigenous capabilities of the shipyard environment of which it is a part. A set of written standards and procedures policed by periodic quality reviews is required to assure continued compliance.

Note should be made that whatever the diagrammatic differences between a phased and structured approach, the greatest difference is the parallel process of test design and program design. Though exceedingly critical and of importance to the life-cycle effort, good software engineering has always held this to be of importance whatever the diagrammatic representation. The point of importance is that a life-cycIe process be defined, and the established use of this life-cycle within the shipyard be used to optimize CAD/CAM integration through the commonality of computer based system attributes, and the use of software tools. Alternative life-cycle representations should, however, be amenable to both business and technical software development/monitoring, and feature a system integration cycle. This form will best serve the coming need to integrate CAD/CAM systems with other shipyard productivity systems, while still utilizing a single software life-cycIe for a given shipyard. The pivotal element of an integration step will aid greatly in identifying, providing and implementing needed software links in the shipyard systems.

5.4.3 LIFE-CYCLE CONCEPTS AND MANAGEMENT

Technical progress within the major activities and the phase and baseline concepts of the life-cycle process represent two different modes of software activities. The technical activities differ from the management activities by the method the demarcation between stages in the life cycle are arrived at. Knowing how to use each stage of the life cycle for both management and technical development is an extremely important tool for cost-effective delivery of software throughout the CAD/CAIM integration process. The development of an understanding of the life-cycle process designations of phase and baseline will enable management to enhance personnel cooperation and the direction and control of system integration and software programming within the bounds of available resources. These interrelationalships are graphically shown on a typical software life cycle in Figure 5.4.2.

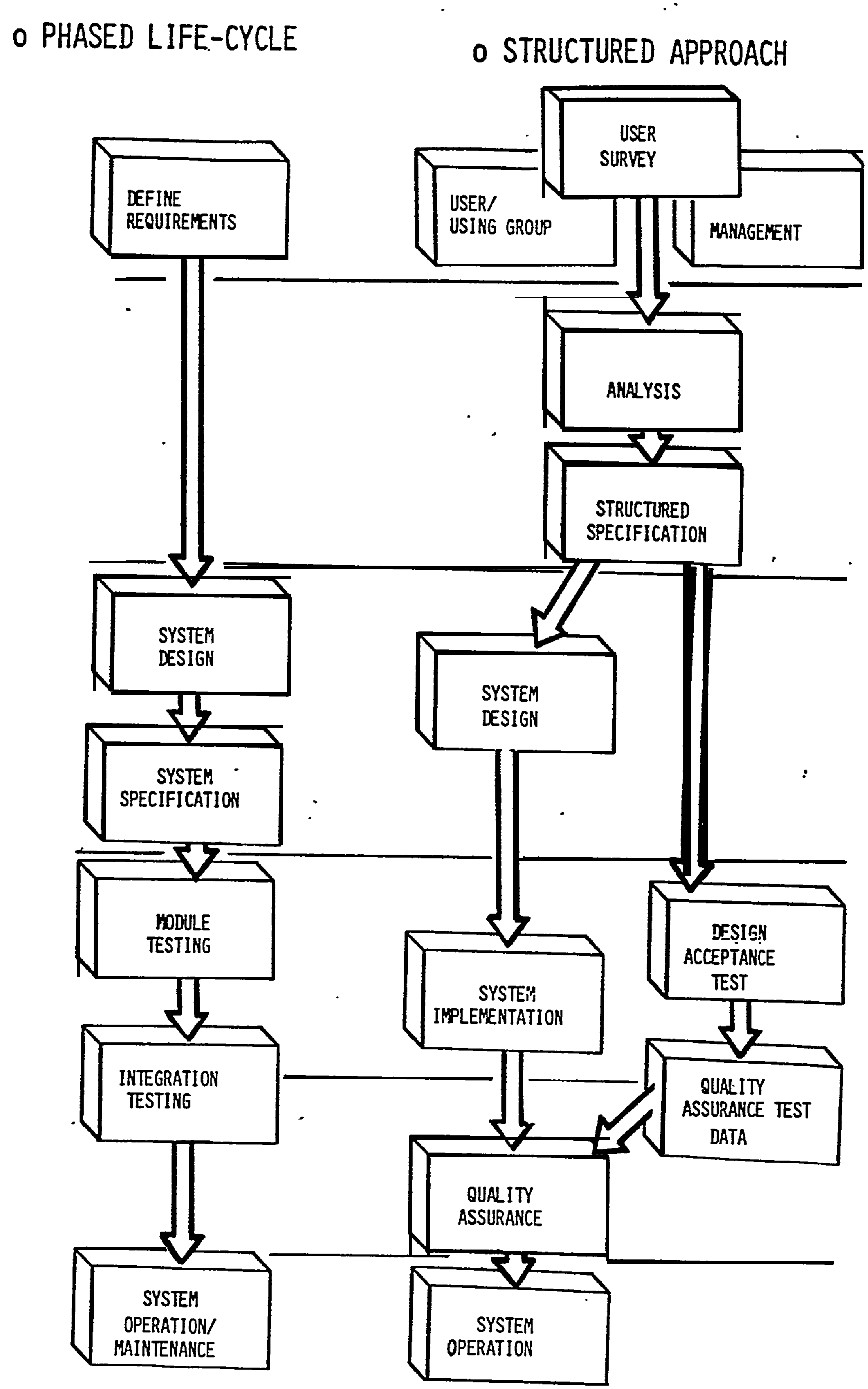


Figure 5.4.2-A

TWO SOFTWARE LIFE-CYCLE APPROACHES.

*The term software Life-Cycle **PHASE** references an incremental slice of time in the devleopment process. The term software Life-Cycle **BASELINE** refers to the reaching of a configuration management point wherein a new degree of control is imposed on software because an assessment of software development technical content shows a defined stage of project maturity has been reached. Software tools are conceptually divided into aids that are technical and managerial based on whether they aid the performance monitoring or reporting of major activities, or phase/baseline sectors of the software life-cycle. Figure 5.4.3 depicts this relationship in simplified form.*

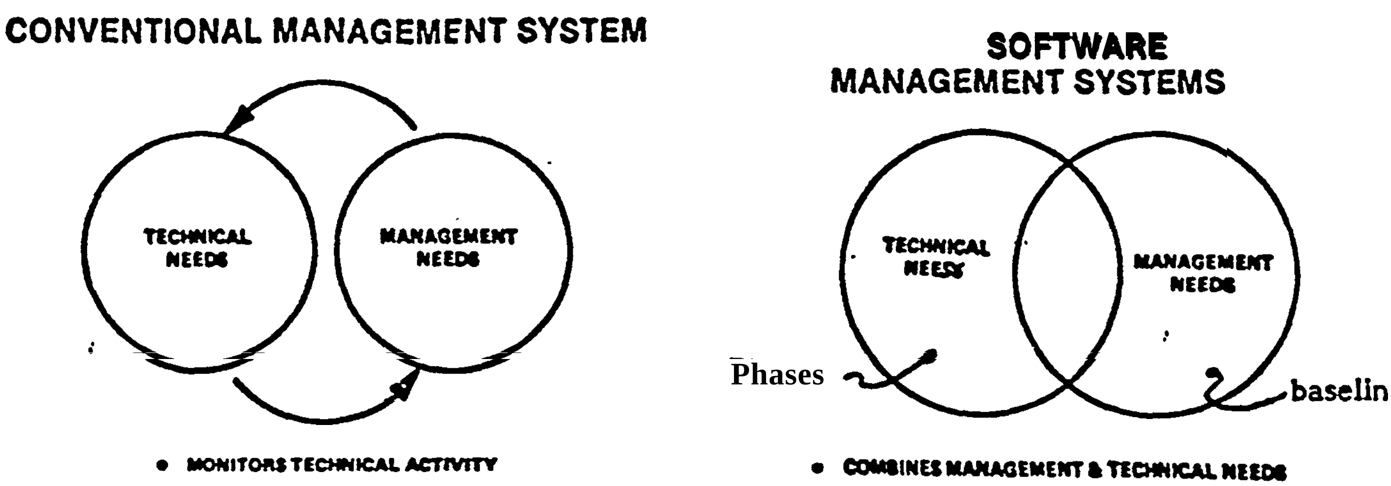


Figure 5.4.3:

RELATIONSHIP OF MANAGEMENT AND TECHNICAL SOFTWARE TOOLS TO THE SOFTWARE LIFE CYCLE

Linking management and Technical Software Tools is important because it builds the software monitoring and control points for the management and technical aspects of a software effort on a single continuum. The software life-cycle process, which is indigenous to the individual shipyard user organization is the foundation on which this system is based. Once this process is affected, software tools can be selected to aid both the technical and management sector, as required, to augment software productivity and integration efficacy.

Referring to Figure 5.4.2, the Four Life-Cycle Phases and Five baselines applicable to each Phase are described as follows:

5.4.3.1 SOFTWARE LIFE CYCLE PHASES

Descriptions given are for the "typical" lifecycle example shown, and can be used as is, or adapted to the needs of the individual user:

1. *Program Initiation.*

   Encompasses definition of mission concept and early system requirements

2. *Demonstration and Validation.*

   Encompasses definition of functional requirements.

3. *Full-Scale Engineering Development.*

   Encompasses detailed design, testing, and integration.

4. *Deploy ment/lMaintenance Phase.*

   Encompasses operation, maintenance, and product improvement.

5.4.3.2 SOFTWARE LIFE CYCLE BASELINES

Descriptions given are for baselines, which are terms delineating groupings of software deliverables. Each of these deliverables becomes a defined list to which a formalized change control procedure is applied, and are thus termed configuration items. Each baseline defines the change from one phase of the life-cycle to the next, and enables both Technical and Management aspects of a program to report progress points reached, and move on to subsequent activities. These baselines are:

1. ***Functional Baseline.***

   Marks end of the program initiation phase and start of the demonstration and validation phase. It is primarily established by the Program Operational Requirements and the System Specification, with baseline control maintained by configuration management reports.

2. ***Allocated Baseline***

Marks end of the demonstration and validation phase. It is established by the Program Performance Specifications and Interface Design Specifications, with baseline control maintained by the Configuration Management function.

3. ***Developmental Baseline***

A dynamic program configuration identification which is initially determined by the Program Design Specification (PDS). The Program Description Document (PDD), Data Base Design Document (DBD), the finaI deliverable version of the program, all descriptive documentation, and the user manuals are also components of the developmental baseline and are added to the baseline as they are approved or accepted. As programs are written and pass minimum acceptance criteria, they shall be added to the developmental baseline under library control. In its final configuration, the developmental baseline shall constitute the software product baseline.

4. ***Product Baseline***

Marks end of development phase. It is established by subsuming the product of the developmental baseline. Product baseline control is maintained by Configuration Management.

5. ***Deployment Baseline***

Occurs in the development phase sometime after Operational Test and Evaluation.

5.4.4 OTHER ASPECTS OF THE LIFE-CYCLE CONCEPT

Only brief mention of other monitoring and control functions of Life-Cycle pIanning can be made here. Three of the most important are the relationship of Hardware/Software, Interface of Programming talent during each phase and the span of control of each major programming skill type.

5.4.4.1: *RELATION OF HARDWARE/SOFTWARE PHASES*

Software phases of system development are integrally associated with the progression of hardware development or acquisition. Figure 5.4.4.1 depicts this relationship.

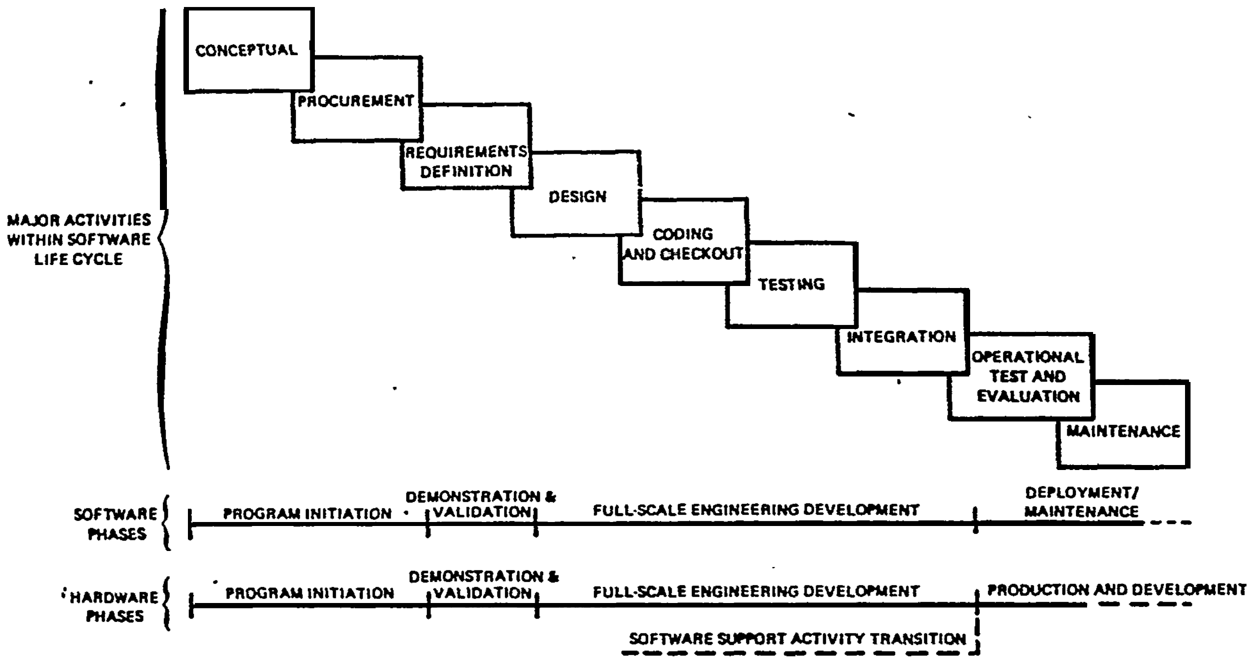


FIGURE 5.4.4.1:

RELATIONSHIP OF HARDWARE AND SOFTWARE LIFE-CYCLE ACTIVITIES

There are notable differences and similarities between Hardware and Software Life-Cycle stages. A major difference is that in the production phase software undergoes extensive maintenance and correction, but does not wear out or require production facilities as does hardware. Note that actual hardware/software coming together takes place during integration where hardware/software undergo integrated test and evaluation (IT&E). At this point software tools report on the status of both software and hardware errors, and track each to assure timely scheduling of further testing. Test Programming Reports and Quality Assurance (QA) Reports are some of these software tool output functions.

5.4.4.2 SKILL ASSIGNMENTS AND THE LIFE-CYCLE

The Life-Cycle of software is totaly different from commonly held concepts of manloading requirements of hardware projects. Since Software is an assemblage of highly diverse, yet very interdependent functions, ail of which are acquired/built on a "First-time" basis, it behaves differently than a hardware assembly task. Figure 5.4.4.2 shows an example of manning rates on a task having little interdependence of all parts (when a piece of work is done, it's no longer a part of future uncertain activities). Compared, in this same figure, are the manning levels for software, where there is a great deal of interdependence between tasks. An example is module testing, where a module tested in the first month of a project must be used again as a prerequisite in some future month to enable testing with another module. This later testing may not work out satisfactorily, thus requiring unexpected re-work of both modules.

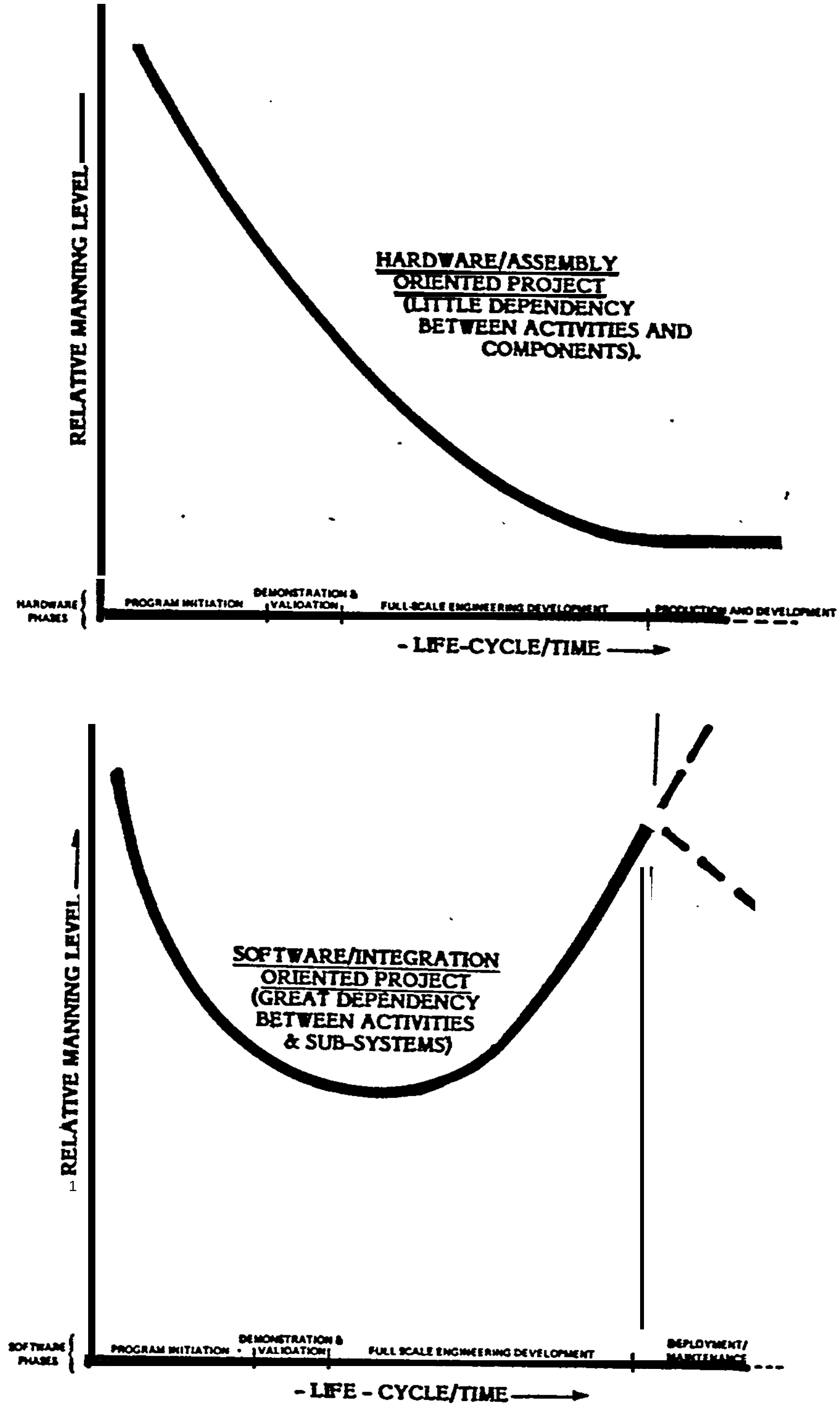


Figure 5.4.4.2:

LIFE-CYCLE COMPARISON OF A HARDWARE/ASSEMBLY AND A SOFTWARE/INTEGRATION PROJECT

The normally expected "learning curve" is seen to reduce levels of activity in connection with the lessening of special purpose tasks, such as jigs/fixtures, for hardware. In the case of Software, this same manning is seen to increase as the totally different, and unknown tasks, associated with system integration take place using as yet incompletely tested sub assemblies of code (modules).

As would be expected, the composition of skill types to cope with the increased need for manpower for Iater stages of software development also varies. Early in a project (initiation/requirements) several system engineers, and a representative programmer and designer are required. Later phases (Development/code) system engineers reduce their presence to a token representative, while several (or many) programmers/designers are required. Later stages (integration/deployment) may see the need for a number of additional system engineers, working with the group of programmers and designers assigned to the project. During the development/maintenance phase this carde of computer specialists can be expected to drop radically in the case of a stand-alone system. However, if the system developed is meant to become integrated with other computer system even more resources may be required. (Note: Broken lines at upper-end of software curve in Figure 5.4.4.2). These computer skills are needed in a different mix over time, use of software tools for each phase must thus be tailored for different users and different missions.

5.4.4.3: LIFE-CYCLE AND SOFTWARE COST

The "Stitch in Time Saves Nine" dictum applies to software development costs due to the same phenomena outlined in Figure 5.4.4.2. Errors, and the difficulty/cost to catch and remedy them is exponentially magnified the later they are caught in the life-cycle. Figure 5.4.4.3 shows this concept. Plotted are the upper and lower

Limits to fix an error, depending on when the error is detected in the life-cycle. This concept is a strong argument for establishing comprehensive requirements, which will lessen the need for change, as well as choosing effective software tools, which will detect errors more quickly and aid in their resolution.

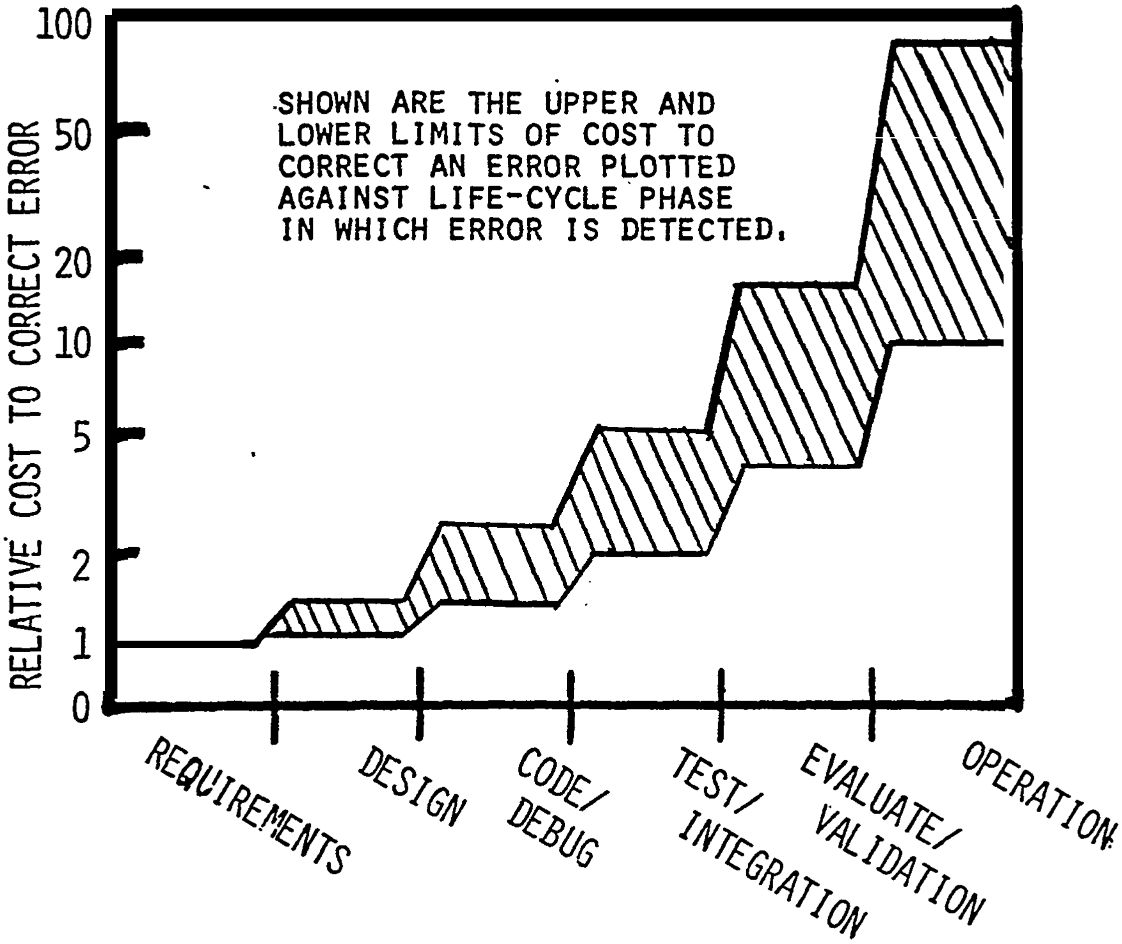


FIGURE 5.4.4.3: COST OF SOFTWARE FIXES AT DIFFERENT TIME OF DETECTION . . . . . *Shown are typical upper and lower limits of cost to correct an error plotted against life-cycIe phase in which error is detected. An important note here is that the careful selection and implementation of software tools will enable a bonus in that their continued use will lessen both the number of errors needing fixes and the efficacy of finding and fixing errors in later phases through automated tracking techniques.*

5.5: TOOLS AND THE KNOWLEDGE TO USE THEM

Software tools are methods, procedures and actual software programs that are used to improve the productivity of software development throughout the software life-cycle. Software engineering skills sufficient to identify the need for these tools are often all that is required to reap large savings for in-house developed, contractor acquired or leased software systems.

Total control of computer software was once the province of selected programmers who ran computers to do tasks required by management. The growth of software complexity has seen the number of programmers, computer skill-types, and project managers grow to the point that it is impossible for a manager to directly interface with a single person who would do the entire task for any given software project that must link with other systems. The ability to perform this *task is made feasible by the use of software tools. Software Engineering skills sufficient to define a shipyard's required software life cycle, and incorporate the use of software tools by both technical and management personnel for all critical life cycle phases can resolve these communication problems. More importantly, software tools can remedy this problem while making possible the design and delivery of large, complex systems not amenable to conventional manual systems development. processes. The CAD/CAM integration process is a system of sufficient complexity to require a disciplined approach to generating system requirements and providing required software. Software tools and their use will become as familiar to shipyard CAD/CAM personneI as were lead designer ducks of the draftsman and lathe dogs of the machinist of just a few years ago. As time progresses, tools change to meet new needs, all that is needed is the knowledge to use them. A basic knowledge of software engineering precepts and the putting in place of a software life cycle indigenous to the shipyard, with supporting standards/procedures documentation, is a starting point. This will enable the selection and use of both manual methods and computer augmented software tools, to permit productive requirements definition through software design installation/test of integrated systems with minimal resources.*

6.0 DEVELOPING SHIPBUILDING NEEDS FOR CAD/CAM INTEGRATION

Emergence of new technologies and procedural methods has radically changed shipbuilding practices in recent years. A primary concern throughout the study has been to relate the complexities of software engineering and systems definition activities to the forthcoming shipbuilding environment that will serve the CAD/CAM integration tasks of the near future. A variety of factors affect, and in turn are affected by, the emerging patterns of new shipbuilding methodologies. A framework within which to define the requirements which define the software needs for shipbuilding CAD/CAM integration is needed. The emergence, wide practice, and proven value of Zone Outfit Planning Methods (ZOPM) has, for all purposes, become the shipbuilding environment of the future. The philosophy that has been developed during this study is that design/production integration will be patterned after operational precepts instituted by ZOPM adaptation in the shipbuilding industry. ZOPM has been, and is continuing to be, developed and has taken on an extraordinary diversity of form for the betterment of ship productivity in many yards.

Our principal recommendation in regard to ZOPM is to stimulate recognition of some of the major opportunities to utilize the planning activity to institute and migrate shipyards to the ZOPM as a starting point to formalize planning for CAD/CAM integration. How this can be accomplished, and where the use of software tools can play its part in augmenting the process is the subject of the next several sections.

6.1 DESIGN PRODUCTION INTEGRATION AND THE SHIPBUILDING PROCESS

Pre-selection of a Strategic shipbuilding perspective, an industry view as opposed to an individual shipyard (traditional) view, has resulted in many differences in the way data have been collected, assimilated, analyzed, and presented for this report. Firstly, the inception of this study emerged from the June 1981 meeting of the SNAME Ship Production Committee, Panel SP-4 on Design Production Integration. The focus of the assembled shipbuilders at this meeting was the formulation of what constituted design/production integration in shipbuilding. The integration of ZOPM, CAD/CAM and ship production phases of Early Ship Design (ESD) and Detail

Design and Construction (DD&C) became a working definition. A simplified concept of this definition of design/production integration is presented in Figure 6.1.

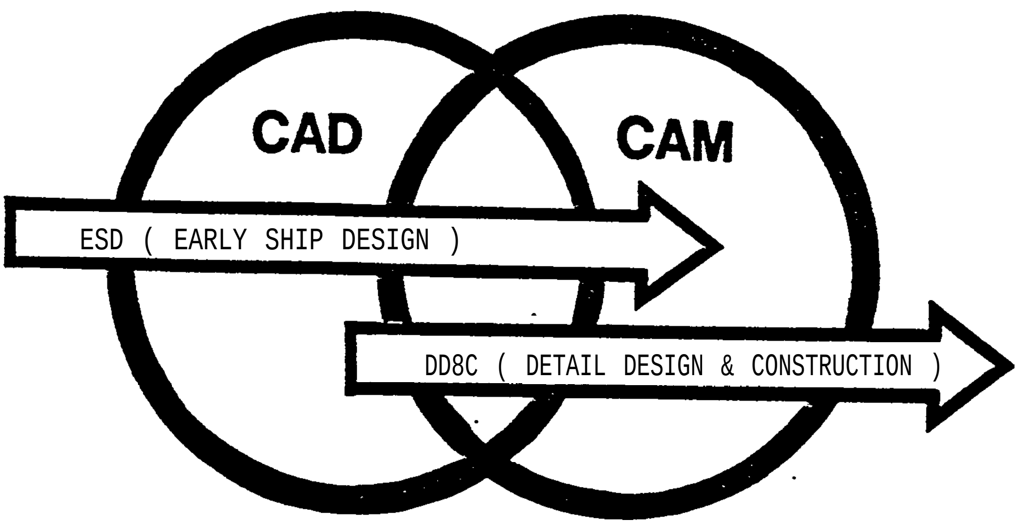
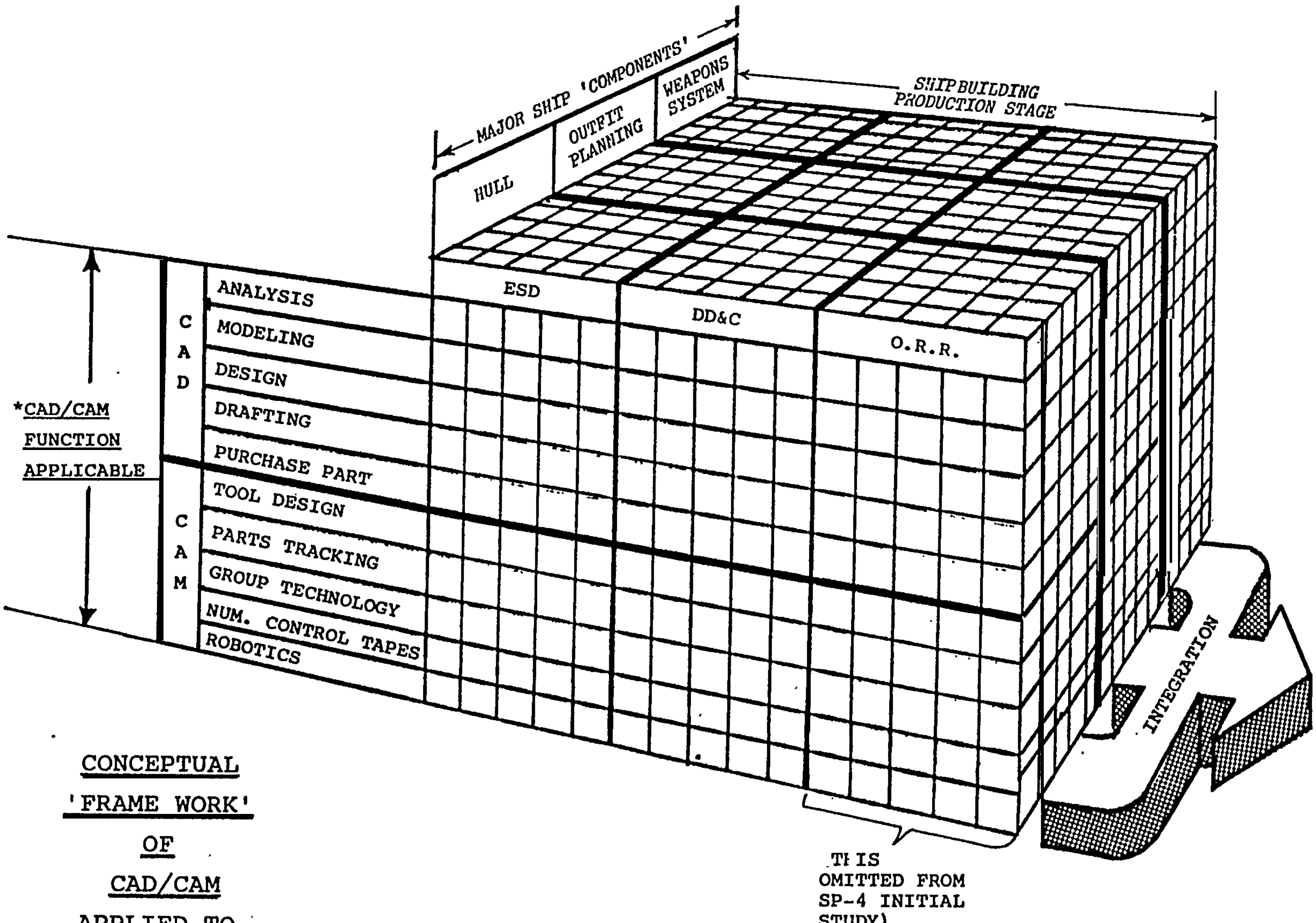


FIGURE 6.1: ***SIMPLIFIED CONCEPT OF DESIGN/PRODUCTION INTEGRATION***

During SP-4 meetings, further development of this concept evolved a chalk-talk figure by the discussion leader which incorporated many of the currently in vogue "Buzz Words" from the CAD/CAM world. Importantly, the graphic concept that resulted was directly related to the design/production integration process of shipbuilding as used on the strategic, or industry level. This cube, depicting the initial working definition of design/production integration, is shown in Figure 6.1-A below:

SP-4
CAM
SUB-COMMITTEE

*(NOTE: INCREMENTAL DESCRIPTIONS ARE INCOMPLETE)



CONCEPTUAL 'FRAME WORK' OF CAD/CAM APPLIED TO SHIPBUILDING

FIGURE 6.1-A: SHIPBUILDING DESIGN PRODUCTION INTEGRATION

The data presented in this figure gives rise to much discussion. A primary focus was the commonality of the computer to both the design (CAD) and manufacturing (CAM) aspects of the expected shipyard of the future. The use of DDS (Data Directory Systems) and DED (Data Element Dictionaries) as aids to software productivity was mentioned and the applicability of these "software tools" to shipbuilding on a strategic level was discussed.

These discussions of software tools as applied to the introduction of CAD/CAM to the shipbuilding process, in the framework of the design/production integration definition, defined the scope of this study. This report is aimed at software tools and the knowledge to use them at the strategic level, or industry-wide ievel, of shipbuilding. At the tactical, or individual shipyard level, this knowledge will enable the selection, evaluation and use of software tools to increase productivity.

6.1.1 Synergy of CAD/CAM Integration Process

Investigations of CAD/CAM integration in other industries quickly isolated the concept of synergy when combining CAD and CAM. This phenomenon, discussed in Section 3, basicaIly follows known system integration precepts wherein the sum of a system output is often greater than the sum of the output of its parts. The most observable of these additions is the inclusion of Management Information Systems (MIS) type systems. Figure 6.1.1 depicts the logic of including these MIS systems.

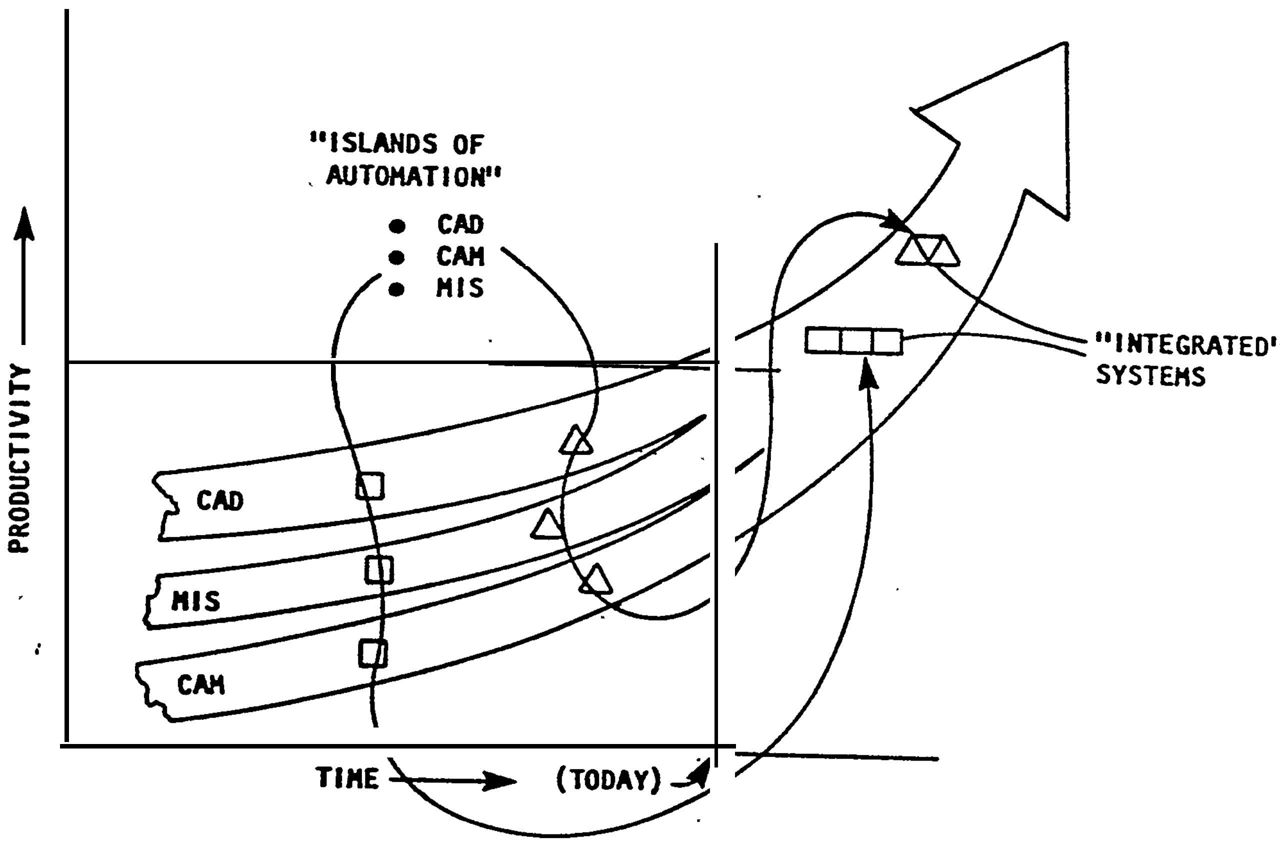


FIGURE 6.1.1 ***INTEGRATION OF CAD/CAM/MIS ISLANDS OF AUTOMATION FOR FUTURE PRODUCTIVITY ENHANCEMENTS***

This inclusion of MIS in the CAD/CAM integration equation is being done in other industries, but is perhaps of even greater importance to shipbuilding.

6.1.2 Assembly Orientation of Shipbuilding

Shipbuilding has a decidedly assembly orientation and is characterized by Iabor intensive operations. These factors are subject to much interpretation, but one generally agreed on conclusion is that Management Information Systems (MIS) types of systems are important to the acquisition of parts, planning of assembly processes, tracking of costs and administration of personnel. Heavy labor intensity hints at great savings through automation of functions. However, a closer examination shows that shipbuilding does not require large numbers of complex machined parts. Thus simply automating machining and assembly shop areas will not greatly improve shipbuilding productivity, even though this is a solution in many manufacturing industries. The real problems are associated with waiting for parts, approved drawings, and instructions for work action. Thus, the focus for CAD/CAM integration in shipbuilding parallels the ZOPM, which takes into account these real-world needs of shipbuilding. Likewise, the planning for use of integrated software systems can cIosely parallel the institution of ZOPM to enable a unified approach to near term, and future introduction of computers to shipyards.

6.2 ZONE OUTFIT PLANNING METHOD (ZOPM)

The current wide practice and development of Zone Outfit Planning originated in Japan. Their technological advances in shipbuilding have largely been made possible by their refinement of ZOPM and related techniques. A brief summary of the Japanese contributions to shipbuilding will show the importance of ZOPM to current day shipbuilding technology, and to the future CAD/CA,M integration efforts in shipbuilding.

6.2.1 Evolution of ZOPM Technology in Japan

Japan presently is the number one producer of ships worldwide in both total tonnage manufactured and total number of ships produced per year. Innovative advances in the shipbuilding processes and employment of technological advances dedicated to the resolution of specific shipbuilding design and manufacturing problems have become their hallmark. Any meaningful. understanding of the metamorphosis of older ship construction methods to current day techniques must *consider the methods employed by Japan, how they were adopted, and where they are going. These concepts have given a meaningful baseline of technologies, nomenclature and methods. from which advanced shipyard techniques can be evolved in America for application to the design and construction of advanced commercial and Naval ships.*

Japanese shipbuilding technologies combine the technical precepts of ship construction methodologies practiced by the Scandinavian shipbuilding industry with the managerial principles of budget, schedule and material control practiced by the American aerospace industry. Japan was quick to adopt these practices in the late 1940's, and by the late 1950's had refined them to a point where their application to shipbuilding was a proven industry practice. Upgrading of shipyards to commence production of the "super tanker" was accomplished by a quantum increase in the technical quality of facilities along with a change in the size of equipment. This investment in new facilities was more than rewarded by the growing share of the world shipbuilding market which Japan was able to capture throughout the 1960's and 1970's. Presently, the focus of Japan appears to be improving the efficiency of yards through upgrade of existing computer capabilities, greater use of CAD and CAM and exploration of robotic applications.

6.2.2 American Technology Amortization

Rapid advancement in Japanese shipyards is perhaps attributable to the readiness of Japanese engineers and industrialists to adapt new technologies, institute requisite skill training and modify management as required to accommodate these changes. This approach was able to enhance overall productivity through the years by a balanced building of technical and managerial capacity at each yard. This contrasts greatly with the United States approach which looks at a technology as a dollar value investment in machinery or a shipyard facility. Under this concept, the dollar value of machinery must be amortized, and when paid for, used as long as possible to accrue a return on investment. This accounting convention concept is tantamount to technological amortization. However, technologies may "wear out" at a time schedule completely independent of "machine amortization[i]. Additionally, the replacement of technologies requires that whole classes of older *techniques/inachinery must be phased out simultaneously, regardless of their position on a cost amortization curve. United States shipyards have not grown as much in their productivity due to the maintaining on-line of older, and in some cases, antiquated facilities. Use of these older equipments promotes adherence to older methods, and often precludes introduction of new productivity techniques even if other elements in the production process. are updated. Adherence to" a practice of "technological amortization" instead of a systems approach to technological update of shipyards, probably best contrasts the American versus Japanese approach to shipbuilding at the beginning of the 1980's. This is certainly not true of all American yards, but even those with conspicuous upgrades of faciIities can neglect the important need to focus on a systems approach to updating technological capability.*

6.2.3 Japan and Quality Shipbuilding

Japanese shipbuilding has emerged as a principle example of their technological prowess. The integration of design and production, recognition of worker attitudes as a production factor, and implementation of material definition as a part of design has led to the rapid building of quality ships. This process is often orchestrated by computer control, and leads to an appreciable contraction of ship delivery times. Additionally, the close degree of control afforded by computer augmented planning enables several different types of ships to be productively worked on simultaneously, even if only a single unit of each is in process. A synopsis of Japanese shipbuilding in 1958, which is very much like the present day United States industry, is contrasted with the ability to effect ship delivery time economies, via computer orchestrated techniques, in Figure 6.2.3. This also depicts the concept of design/production integration, where the integration is accomplished by including material requirements planning and procurement earlier in the linking of design/production cycles. .

A comparison of conventional to the zone outfit planning method (ZOPM) is shown. In the ZOPM note the overlap of outfit design, material definition, procurement, and production which has been achieved by (Japanese) shipbuilders. Through this method when only 30% of a design is completed, 70% of its required material is defined.

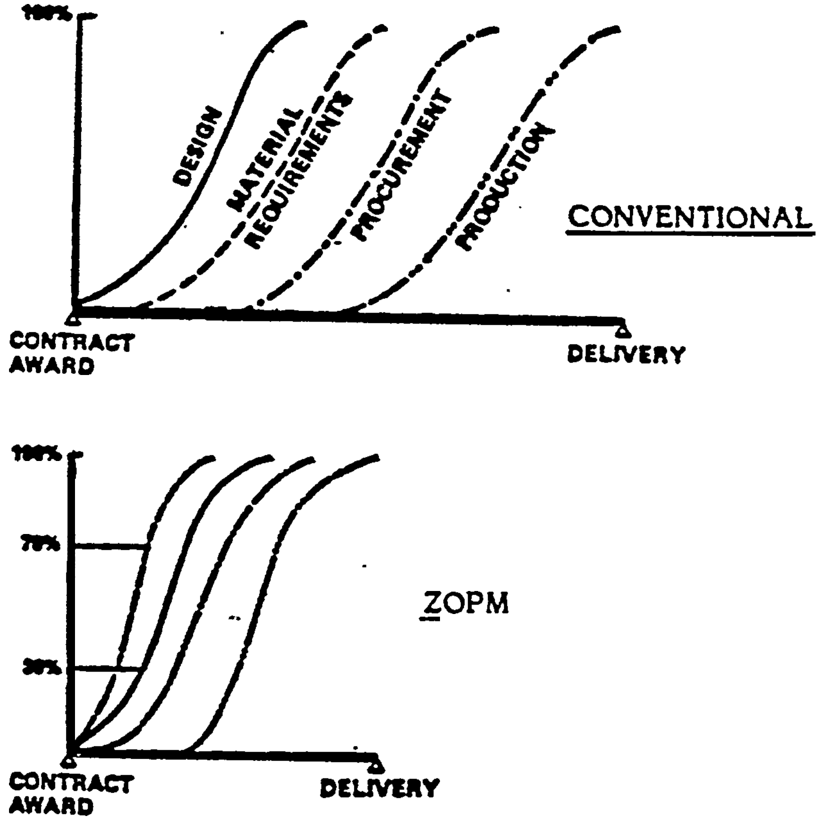
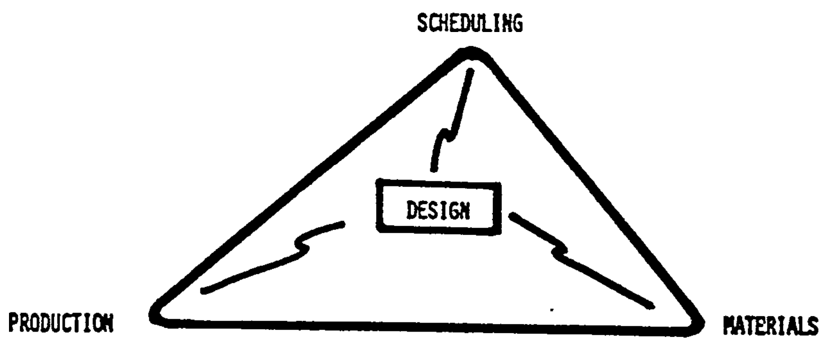


Figure 6.2.3: ***CONVENTIONAL AND ZOPM SHIPBUILDING***

Concentration on employment of standardization and focusing on design as a pivotal element in the shipbuilding process has enabled this success to occur. Automation of the functions of production, scheduling and materials are drivers of design activities to enable these accomplishments. A very positive approach to utilizing computer aided design has evolved a complex of automated design and design support functions. Success in utilizing these computer based techniques has been transferred to varied ship designs. The design functions drive other data bases, speeding up deliveries and collection on completed contracts. Outfit planning is the key technique enabling the achievement of their success in shipbuilding. Figure 6.2.3-A shows the relationship of these factors, with design as a pivotal element.



JAPAN'S APPROACH TO COMPUTER AUTOMATION OF SHIPYARD:

- THEY ASK "HOW TO USE CAD", NOT "SHOULD WE ..."
- USE THEIR SUCCESS IN TECHNOLOGIES TO GUIDE MARKET SURVEYS
- DESIGN FUNCTION DRIVES OTHER DATA BASES
- OUTFIT PLANNING STRESSED

FIGURE 6.2.3-A PIVOTAL ROLE OF DESIGN IN JAPANESE SHIPBUILDING

6.2.4 OUTFIT PLANNING

Outfit planning is a methodology of ship construction that is concerned with production activities for other than the hull components of a ship. Traditional treatment has relegated outfitting activities as a successor function to steel planning, an activity following hull planning in a serial fashion. A recent emphasis, *originated in Japan, is that production planning should integrate both hull block construction and outfit planning. Essentially this integrated concept enables outfit planning to be accomplished towards the most efficient utilization of men, materials, tools and techniques to "stuff" individual hull blocks before assembly. This enables economies of time, materials and cost heretofore not possible in shipyards where both design and installation was accomplished on a system basis (i.e. piping, electrical, etc.). Modern zone outfit planning methods (ZOPM) initiates functional design and planning by system and transitions these to details for zone design drawings. This allows interference checking, production planning, and the ordering of material by zone, as well as fabrication by zone and stage to accrue construction economies. Although Japanese shipyards have advanced hull block construction methods (HBCIM) beyond the state of technologies practiced in the United States, it is the perfecting of zone outfit planning methods (ZOPM) which has given a decided competitive edge to Japanese shipbuilders. Thus the process of outfit planning, how it is augmented by currently in-place and projected computer techniques, and, benefits attainable will be the focus of the following sections. Outfit planning is the single most important concept on which the future growth of American shipyards and CAD/CAM integration will depend.*

Outfitting of a ship has historically been accomplished on a system by system basis. This procedure necessitated the design of a complete system, such as hydraulic lines, its relationship to the completed ship, and a procedure. to fabricate and install the system parts in the proper sequence in the completed ship. Outfitting is then accomplished on a system-by-system-basis. A very visible problem associated with this method has been the great numbers of workers from many different trades, who, along with their associated special tools, must work together in the cramped quarters of a crowded ship hull to complete installation of the outfitting task. This causes much wasted motion and delays due to the sheer interference of personnel. Because of these factors, shipbuilding is the most labor intensive industry in the United States, surpassing even the construction industry.

6.2.5 Zone Outfit Planning Method (ZOPM): Application/Approach

Japanese shipbuilders are circumventing the problems associated with conventional outfitting by preoutfitting. This method applies resources earlier by outfitting large structural sections prior to erection of a hull. This necessitates construction of steel assemblies in a sequence which is not optimum for maximizing steel throughput with minimum expenditure of resources. Preoutfitting also requires dedication of appreciable time and facilities, e.g., large indoor or outside areas where access is improved but components are stiU installed piece-by-piece using conventional methods. There is great dependence on management of the production process because if a hull section is not available, outfitting is disrupted.

Preoutfitting is planned by allocating resources to activities associated with ships' systems. Access is improved depending on the sizes of hull blocks, but craftsmen still compete for time and space. Getting tools and materials to the work site is still not idealized, but there is some improved ability to level load the outfitting trades thus improving productivity by permitting more uniform work flow. But savings in total manhours and the overall building period are inherently limited because the only real distinction between preoutfitting and conventional outfitting is where the work takes place. In practice preoutfitting a very large structural assembly is the equivalent of outfitting a small ship of equal tonnage by conventional methods.

Zone outfitting addresses everything within a limited 3-dimensional space, thus going one step further. It frees outfitting as much as possible from dependence on hull construction progress and from arbitrary resource demands required by installation by system. These ends are achieved by addressing certain interim products, i.e., significant subassemblies of just outfit materials that have been joined together away from a hull errection site or an outfit pier. This added degree of freedom permits segmentation of a production process by classes of problems so that common solutions can be applied regardless of both product configurations and where outfit components belong in ships' systems. This is a principle of Group Technology which is a still developing industrial science, invoked to some degree everywhere for hull construction but yet to be applied by most shipyards for outfitting.

Zone outfitting is analogus to the Hull Block Construction Method (HBCM) which has been highly developed by shipbuilders throughout the world during the last three decades. In fact, the HBCM is a prerequisite for zone outfitting. The HBCM employs zone-by-zone construction as compared to the archaic system-by-system method. The singular advantage of zone-by-zone construction is that it permits more freedom in applying a logical work breakdown structure to achieve more uniform production flow. But while the HBCM has been almost universally adopted, the same logic for outfitting is not yet in general use. When the same logic is recognized there will be acceptance of the premise that outfitting is not a successor function and that it is necessary to plan and build a ship in a manner which will allow outfitting and hull construction to be accomplished simultaneously.

6.2.6 Identified Stages of Zone Outfitting

Three major-areas of production are identified in outfit planning:

- ***On Unit** production of sub-assemblies of outfit materiais only, away from any hull components. An example would be the creation of a series of electrical control panels meant for later installation on board an assembled ship, or a particular hull block.. This concept facilitates letting out work for bids by outside sub-contractors and allows the ready application of group technology principles.*

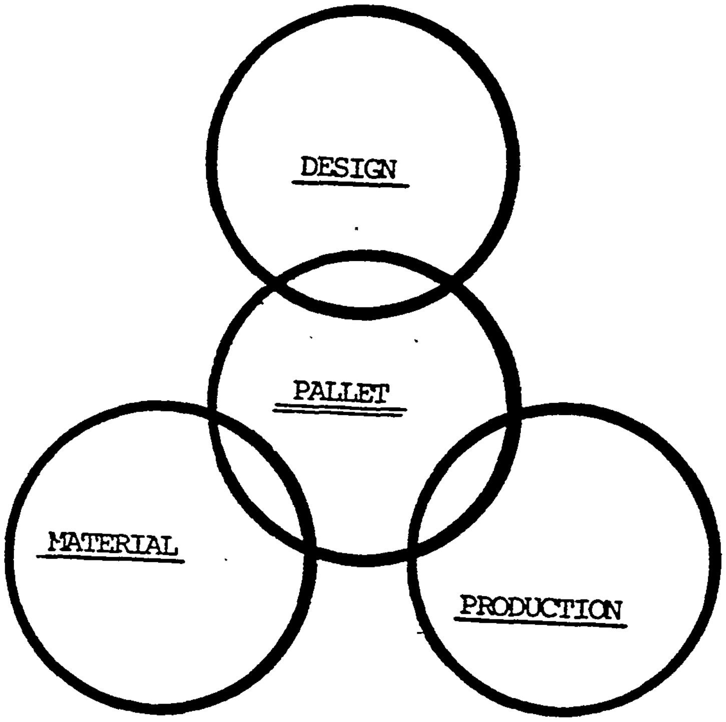
- ***On-Block:** Installation of components, or completed on-unit assemblies, onto a hull block. Planning process sequences these operations to enable assembly at time when benefits accrued would be greatest, such as the installation of control panels when upper deck is still off to facilitate work effort.*

- ***On-Board:** This phase is limited to connection of units and blocks into the integrated ship configuration, including subsequent inspection, painting and tests.*

Simplistic as these concepts sound, they enable, through a host of supporting techniques and procedures, greatly increased shipbuilding economy and efficiency.

6.2.7 *Information Organization Requirements*

Outfit planning is necessarily supported by a well organized flow of information. Concepts of. information handling foreign to earlier shipbuilding methods have emerged. An important tool is the pallet concept. The pallet indicates a work package in construction. It designates a group of materials, and enables staging of material for kitting and delivery to a work site. The use of kits ensures that a worker will have all of the components needed to do a job. Figure 6.2.7 shows the pallet as the information link to define a stage of construction.



"PALLET" defines a zone at *a stage of construction in design, to indicate a work package in production, together with its designated group of materials to accomplish work. Used in this way the Pallet becomes the "information link" to integrate design-production and MIS Systems.*

FIGURE 6.2.7: PALLET CONCEPT FOR ZONE OUTFITTING

- *Flow of Information in Design:*

  Receipt of components on time becomes critical in the ZOPM. This is so because a missing piece from a pallet when it is needed can negate any accrued savings from the use of outfit planning techniques. Because of this, it is often better to purchase components at a premium, rather than risk late arrivals of parts and pay a penalty in labor. Thus, design accelerates purchasing. This capability is aided by parts standardization, where the ability to purchase early, in quantity savings, can expedite the design process. Where possible, the contracting of large sub-assemblies also facilitates construction. Thus, the use of ZOPM is seen to depend heavily on the flow of information in design, where early identification of purchase parts, production methods, and pallet needs accelerates the ship design-build cycle at a cost savings.

- *Planning and Scheduling Simplification:*

  Upon receipt of a contract a shipyard must institute planning activities. Determination of the block size, zones and major units must be made. Shipyard facilities must be committed to accommodate the production of the ship and resources committed to specific calendar dates.

  Separation of the production process into hull construction and outfitting enables detail planning for each to proceed in parallel until such time as they must merge. The techniques of on-unit, on-block and on-board outfitting permit the avoidance of details, enabling a concentration on large interim products. The economy in the planning process is achieved by rapid organization of information to describe interim products, thus reducing the total volume of data required.

6.2.8 Simplication of Ship Production Processes

The ZOPM has evolved several operant practices related to the production process. Two of these are to:

(1) Simplify assembly methods

(2) Transfer need to understand assembly techniques from production to design.

Both of these are critical to the design/production integration cycle. A discussion of these follows:

- o *Simplification of assembly methods: The use of the pallet concept aids greatly in actual manufacturing of the ship. Its role in ship construction as an information link with design, material and production needs is presented in Figure 6.2.7.*

- 0 *Systematize the design production process: A related concept utilizes "Patterns and Panels" to systematize and accelerate the design/production process. This concept utilizes the fact that many system diagrams reflect patterns that are similar, even if for a different size and type of ships. This is especially true when compared for specified zones. Panels can be identified, and compared to similar panel descriptions in shipyard files. A panel listing so located gives data on: required standard fittings (quantities, descriptions, etc.); related standard materials required (pipe, wire, etc.); and, standard design/production guidelines for materials utilized. Identified panels selected are adjusted to a specific design by functional engineers who:*

  Modify standard (panel supplied) guidelines per shipbuilding specifications.

  Add fitting sizes

  Prepare list of sizes and estimated quantities of non-standard materials.

- o *Transfer the need to understand assembly techniques from production to design: This is, perhaps, the pivotal concept in ZOPM. Essentially the goal is to understand the practical "how to accomplish assembly!' problem uniquely presented for every ship as early as possible and in as much detail as possible. This enables early specification of assembly sequences, etc. One key method is the use of enhanced communications. This is in keeping with the premise that information is a resource used to track material, design, production, etc. Formal meetings are* *held by inter-disciplinary skill groups* to *make decisions for action. The meetings themselves are treated as milestones, with specific outputs required.*

Other methods to bring assembly technique understanding to the design stage includes use of automated assembly trees and use of three dimensional models to serve as a basis for photogrammetric creation of computer based working drawings. The basic goal is to direct workers to an achievable completion of a defined work task by integrating design and fabrication 'smarts' as early as possible to enable rapid acquisition of parts and scheduling of facilities.

- *Material Control - requisition and delivery of material to production: A key element of outfit planning is the early ordering of standard parts. This function must be mated with parts storage implementation which creates pallets as expeditiously as possible. Assemblies are scheduled by pallets by top-down scheduling techniques, which must be coordinated with hull milestones. Essentially, the flow of material and information must be mated to create a smooth transition of design, received parts and production by the ship outfitters.*

6.2.9 <u>ZOPM: Advantages Accrued</u>

Purely mechanistic advantages of zone outfitting have been discussed. However great the advantages of schedule, resource utilization and cost savings are, it does not tell the complete story of benefits accrued to the shipyard employing this method as developed in Japan. Zone Outfit Planning tangential benefits include its ability to:

- *Reduce the outfitting period*
- *Minimize outfitting on-board*
- *Simplify outfit planning*
- *Avoid interferences between trades*
- *Achieve greater efficiency of erection cranes*
- *Reduce overhead work*

- *Improve facility utilization*
- *Improve Safety*
- *Improve Working Environment*
- *Increase productivity*
- *Improve Quality*

This integrated view of ZOPM benefits shows that many tangential benefits are inherent in the methods empIoyed. Noteable are improved worker safety, better environmental conditions and general improvement of the worker's motivation. Workers in Japan who perform tasks using the pallet concept of the ZOPM must become more versatile, as they must often be accomplished at several different skills to complete a task. However, this challenge to cross train has added an element of interest to the shipbuilding industry, and motivated the individual worker to new productivity heights. Thus any widespread adaptation of the ZOP M by America must recognize the role of the individual worker before instituting purely mechanistic changes. The problems which arise are due to a worker being asked to step outside his normal craft to do work in a zone rather than apply his designated skill to a system. Resolving these problems must be a part of the CAD/CAM integration activity in shipbuilding.

6.3 SHIPYARD PRODUCTION ENHANCEMENT FACTORS

Investigation of methodologies employed by advanced Japanese shipbuilders by other National Shipbuilding Research Program studies has identified factors which enhance productivity. Factors in the zone outfitting method are the employment of group technology, emphasis on communications between skills, combining of design/production functions, stress on motivational programs for all employees, and the formalization of the technological innovation/transfer prwess. It is important to note that the success of instituting these factors has occurred largely without complex computerization. Computer systems are now being used to integrate all aspects of shipyard design, production and management. Typically Japanese shipbuilders ask not "should" computer automation be applied to shipbuilding, but rather they ask "how" to apply computer automation to shipbuilding. Computer usage enhances production by using computer techniques centered in the design *phase to drive production, materials and sdheduling functions. The design function drives other data bases, with outfit planning stressed. " Success in adaptation of new technologies is used to guide market surveys to gain new footholds in the world shipbuilding market. Current application of computers follows a well thought out, comprehensive shipyard modernization plan.*

6.3.1 Shipyard Computer Applications

Use of computers has become a part of shipyard production enhancement. This is so because modern shipbuilding/outfitting techniques, especially as developed in Japan, are more amenable to computerization then older construction methods. A second reason, unrelated to shipbuilding, is that the availablitiy of computer aided design, manufacturing and robotic technology has matured to the point where applications to shipbuilding have been proven. Computer application to virtually all aspects of shipbuilding is now a task being accomplished in parallel with the updating of shipbuilding methodology.

6.3.2 Use of Computers: CAD/CAM

Japan has actively utilized European, Scandinavian, American and their own computer techniques for shipbuilding since the 1950's. Use of computers has been prevalent to such a degree that a working CAD/CAM thesis has been formulated for future development. This thesis states that the most productive use of CAD/CAM will be that computer system that effectively integrates CAD/CAM, one with the other and both of them with other computer data bases of importance to HBCM/ZOPM systems. This thesis has given a new meaning to CAD/CAM and certainly a new scope to the challenge of creating computer systems for shipbuilding. Figure 6.3.2 compares a conventional view of CAD/CAM, where the focus of the conventional approach is on design/drawing automation coupled with numerical controlled metal bending/cutting. A contrast is seen in the emerging shipbuilding view, largely made possible by ZOPM maturation, wherein the computer and use of relational data bases from material, scheduling, shop methods, etc. are involved to aid the shipbuilding process. Use of an integrated CAD/CAM approach enables economies that justify computer applications which may not be possible if computers were applied only to design (CAD) functions. Active interest in

CAD/CAM integration, tailored to shipbuilding, is a current area of activity and concern in both Japan and the United States.

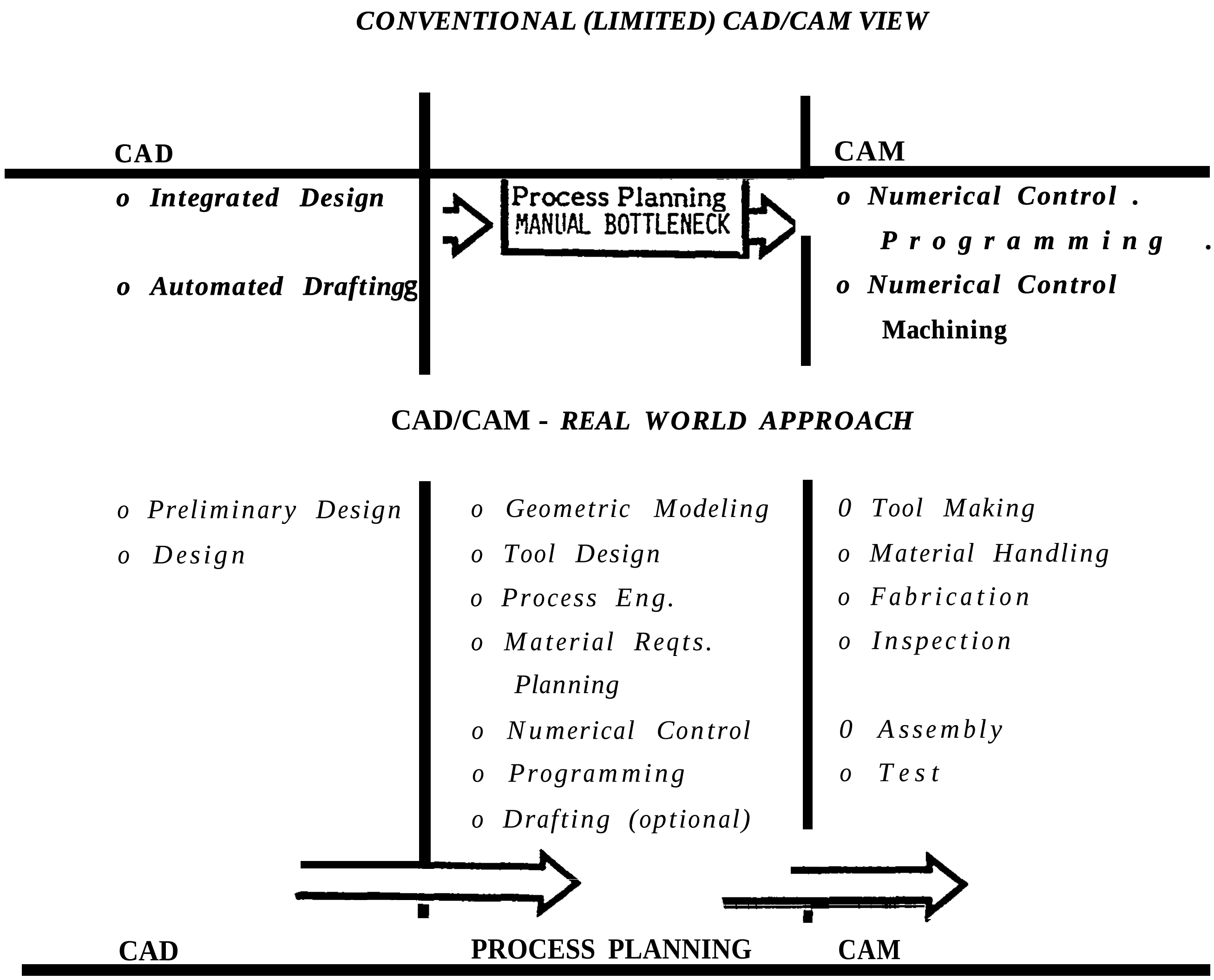


FIGURE 6.3.2 ***CONVENTIONAL AND INTEGRATED VIEW OF CAD/CAM***

In line with the CAD/CAM synergy effect caused by Design/Production integration, integration of CAD/CAM results in other tasks being affected.

The concept presented in this figure on Integrated CAD/CAM is more real world then the conventional view. However, when combined with CAD/CAM scenarios *from other industries, using new technologies, this integrated view becomes even larger in scope.*

6.3.3 Use of Computers: MIS and Process Planning

Zone outfit planning procedures permeate virtuaIly all aspects of shipbuilding. Since shipbuilding systemization, via ZOPM, involves. the coordination of hundreds of thousands of discrete parts, assemblies and processes, proficient planning is a critical factor in successfully utilizing these methods. Only computer manipulation of these functions can cope with the data/information requirements needed. For want of better terms, Management Information Systems (MIS) and process planning are the terms applied to this grouping of computer functions. Japanese shipbuilding practices have aptly demonstrated the practicality of using computerized large-scale parts tracking, inventory, planning and scheduling programs. The ZOPM enables a practical Work Break-Down Structure (WBS) to be formulated, which in turn yields the fundamental elements for a comprehensive data base of materials, operations, costing and scheduling of each individual ship. Concise specifications of available shipyard equipments, skills and facilities, similarly committed to an accessible data base format, enables simulation of the interaction of the shipyard and planned ship to yield optimal resource commitment for a planned project. This also enables cost savings through early recognition of applicable parts standardization. Thus, MIS and supporting process planning functions, are melded with the discipline of the ZOPM and logical WBS of each ship to enable Japanese shipyards to use the leverage of computer power to enhance productivity. With the advent of CAD/CAM, an important unresolved question is the degree to which MIS functions will become an integral part of the automated design/manufacturing functions. Further applications of computers in this area must be patterned after the rigorous, ship oriented, planning methodologies as represented by ZOPM and become a part of the CAD/CAM integration process.

6.4 SYSTEMS INTEGRATION NEEDS

A primary concern throughout all phases of the CAD/.CAM system integration effort is the development of a means to relate the technical aspects of computer hardware, software, and communication networks to the mission they were intended to accomplish to aid shipbuilding productivity. To facilitate this, the initial systems definition process should evolve into a set of written documents which spell out the functional requirements of the system. The functional requirements form a foundation on which to build the hardware, software and system test activities. Once the goal of written requirements is accomplished, the task of integrating CAD/CAM systems into its operational environment can commence. Concern with both the managerial and the technical problems arising in connection with integrating CAD/CA,M into a shipyard environment embody the scoping of the systems integration effort for what is a large, real-time computer project.

6.4.1 Technical vs.Management Aspects of System Integration

A complex system, such as a shipbuilding CAD/CAM system, requires the identification and organization of many tasks. These tasks are highly complex in nature, and a decision to focus on either the managerial or technical aspects of an identified series of tasks must often be made. This decision being made, the mix of project tasks continues toward completion, often with great initial success. However, after a percent of time, perhaps as late as 50 percent through a project schedule, a problem in matching the interfaces of tasks emerges, and system integration of otherwise working and completed tasks cannot be accomplished. This problem is manifest by the following symptoms:

- *Mismatched interfaces of debugged modules in software integration*
- *Mismatched data and control flow in tested subsystems*
- *Mismatched hardware/software interfaces in hardware/software integration phases*
- *Mismatched - management terms and reports relative to both problems and solutions - thus totalIy incapacitating a project.*

A primary reason for these, and a myriad of related problems in system integration involving computer hardware and software, arise from the well-meaning progression of uncoordinated technical and managerial efforts. The only cost effective remedy, found through years of system integration experience, is pIanning for total prevention of this problem. There is a dichotomy of technology and management aspects of any computer project. This disparity grows exponentialy as a project grows in sheer size, diversity of functions, complexity of networks, and operating speed. Imposition of a monolithic centralized control during the brief span of a project is an impossibility, thus a high degree of systems integration expertise is essential to enable the successful subsystem developments, in the development mode, to become successfully operating components of the integrated system. Extensive experience is a prerequisite to provide the required level of systems integration expertise to cope with the systems requirements formulation and formidable integrating problems which are characteristic of CAD/CAIM integration projects.

6.4.2 Formalized Approach to Systems Integration

Recognition of the nature. of systems integration problems, and years of experience in development of both scientific and business data processing systems, has resulted in a mature approach to dealing with the issues raised. An important first step is to have the required staff, facilities, and tools to implement this formal approach for CAD/CAM integration. Required are the following key features:

- *Development of a comprehensive System Integration Plan (SIP)*
- *Attention to Interface Controls*
- *Establishment of a System Integration Team*

An abbreviated description of each of these features of CAD/CAM systems integration and the role software tools play follows.

6.4.3 System Integration Plan (SIP)

The SIP (System Integration Plan) is a document that is prepared at the start of a program, reviewed on a regular basis, and reviewed as required to effect a smooth *system effort throughout a program. Importantly, this document includes both technical and managerial aspects of each software and/or hardware subsystem (intra-module relationships) aswell as similar data between each subsystem (inter-module relationships). This plan details required integration needs so that important questions can be addressed by management well before the integration phase. Typical points that are addressed by management and planned to avoid later problems using the SIP are:*

- *Is integration manpower loading suitable to the task? (training, number, availability)*
- *Are required software tools available to initiate and complete all aspects of integration testing?*
- *Will a special hardware/software integration facility be required?*
- *Has sufficient time/skills been allocated to develop test plans?*
- *Are all aspects of test/verification/validation procedures of integrated systems complete?*
- *Will complete documentation be available to facilitate integration?*
- *Can the change control system adequately handle problem resolution/re-test during integration?*

These are just a few of the typical questions that are resolved by a suitable SIP. A well-developed SIP avoids major problems during integration. Some of these problems relate to incorrect integration of schedule estimates, uncontrolled change of functional requirements, lack of problem/configuration control or mismatches of hardware/software changes during development. Importantly, surprises due to a lack of integration tools, computer/systems, and special facilities can be avoided by a properly developed SIP. The published SIP document will form an important part of the management/technical reporting tools for the duration of a CAD/CAM integration project.

6.4.4 *Importance of Interface Controls*

Experience has shown that successful development of subsystems can be achieved, given the proper skill and time allocations. However, the internalized generation of these ,many "mini-processes" within the overaIl CAD/CAM system will have *little bearing on the needs for their interface with other subsystems unless care to provide a project-wide interface control is taken. The interface controI aspects of system integration on many projects over the years have shown that interface problems become especially cumbersome in networking-computer-communication-data base systems. A comprehensive solution, requiring disciplined project management, is early institution and policing of a project-wide Interface Control Documentation (ICD) system. The use of ICDs is, simply stated, the identification of all interface points essential to system integration via the SIP, and requirements for a specific document, acknowledged and reviewed by all parties, describing the needs of the interface. Use of ICDs enables constant formal management review of integration needs before they arise. The ability to categorize integration problems, into groups by similar characteristics will permit dealing with problems in "batches". This will enable dealing with "batches" of similar problems at one time, permitting efficient use of scarce special skills to examine or solve problems. Streamlining of the integration process <u>before</u> critical software testing or hardware/software integration is thus made possible. These reviews include complete management approval and changes to the SIP. Virtually all ICD activities are covered to some degree by in-place test, configuration or programming documentation, but the ICD enables specific focus on integration needs and omissions in other systems to be highlighted. The ICD update system is the principle means recommended to keep the SIP current during a shipbuilding CAD/CAM integration project.*

6.4.5 Establishment of a System Integration Team

A well thought out SIP, mated with an in-place working system of ICDS, cannot alone provide for effective system integration for a CAD/CAM integration project. A means to go beyond the planning of system integration and reporting essential interfaces must be in-place at project start to insure that plans are controlled. This can be insured by development of a system integration team, whose responsibilities for large systems must run the gambit of interdisciplinary hardware, software, "peopleware", and managerial skills. This team is often comprised of part-time attendees from on-going project tasks in smaller systems. However, it must always have a single, identified Integration Director vested with sole *responsibility and authority to provide review, guidance, and control of the system integration process. The helm of this special team requires that control be exercised over at least these aspects of a system:*

- *System configuration control, and knowledge/review/participation in the change process.*
- *System integration team control, including staffing, organization, and direction with in-place corrective procedures to control perturbations to the system configuration during development and/or activity disparities during the actual integration phase.*
- *System schedules, including those development activites required by the SIP prior to integration to insure knowledge of programs and realistic resource interests.*
- *System standards, development, review, and implementation so that integration team members can be appraised of project test and operational missions.*
- *System software and management tools to enable informed review; this includes a role in selection/specification of such tools toward the end of facilitating system integration.*

Experience has shown that a system integration team director is a keystone in the development of a system as well as its actual integration. This is so because he must develop his team into an elite group of specialists conversant in all project subsystems, technologies, tools, techniques, and personnel. The System Integration Director thus becomes the standard bearer of the need for a completed system, resolving technical complexities into the single goal of an integrated, smoothly working system. This image aids in enabling special-purpose ad-hoc committes to be formed, as required, to solve special problems. Mixtures of these ad-hoc committees can be drawn from available in-house software, hardware, communications, or even vendor populations. Experience in orchestrating the vital aspect of system integration shows the critical importance of early instituting of this integrating team effort in conjunction with personnel involved with CAD/CALM development.

6.4.6 The System Integration Process

Experience with system integration must lie in both highly complex, large-scale technical and business systems. The ability to integrate complex hardware, state-of-the-art software, extensive voice/telemetry interaction, and networking problems is critical to large integrated systems. However, a critical point of understanding is required to qualify these statements. It should be realized that system integration difficulties in commercial systems can be greater in scope then most of the scientific systems applications. Scientific systems can be programmed and tested in a series of fits and starts. When they are in use, temporary down time conditions are tolerable. However, business systems must be programmed, tested, verified, and operational prior to completion of a point where any useable work can be accomplished. For this reason, system integration as a focal point of managerial/technical intercommunication throughout the integration project must be a driving function. A comprehensive expansion of the techniques presented herein to accomplish this integrating mission is a minimal requirement for effective CAD/CAM integration.

6.4.7 *System Engineering Tools and Methods*

A primary concern throughout development of a project will be the relating of complex system elements such as communication networks, computer hardware/software, facilities and personnel to meaningful managerial evaluation, control and verification steps. These system elements are affected by a variety of developmental factors, and the mode of development of any system elements will, in turn, affect other system estimates. A philosophy developed during many years of computer system development, strongly influenced by the diversity of problems to be expected in large CAD/CAM system development similar to shipbuilding projects, has been to utiIize a system engineering approach to projects. System elements must be developed, integrated, and managed within a total systems framework. A critical requirement is that the management of all system elements within a system be accomplished consistently within a framework of policies and procedures that are clearly set forth in unambiguous form. Implementation of these standards and procedures must consider the type of system element being managed. Attention must be given to both technical and administrative considera-

tions that affect costs, feasibility, schedule, project/personnel efficiency, and design/user relationships. Systems Engineering is the recognition of the above stated nature of system elements, the identification of technical development steps throughout a projects life cycle, and the organization of these steps into a manageable flow of activities.

Analysis and design of an integrated CAD/CAM system and its system elements will occur at a series of levels. Requirements which now exist in the form of general objectives, to be met by the system, will be differentiated and refined. A first step will be identification of functions, on a gross scale, which the system must perform to accomplish the general CAD/CAM mission objectives. Such performance requirements as speeds, capacities, design constraints, and requirements essential for system support, development and test, are also identified. Subsequent steps analyze functional requirements through progressively lower levels, where functions are allocated at each stage. System elements are decomposed to system segments, then to configuration items (software or hardware) and finally to a module level. Systematically carried out at all levels, this functional approach maintains visibility of each element in the system as .a whole, clarifies interrelations between elements, and highlights the need to investigate possibIe missing eIements. The systems engineering process is resolved into a standard flow which identifies technical milestones and sequence of accomplishments. Each major decision point can then be easily identified, and management milestones are inserted to provision both visibility and control for the development activity. Close attention to details, which otherwise would be overlooked, becomes possible using this system engineering approach. Using this system engineering process also insures that system elements will be developed systematically via inclusion and scheduling in the SIP.

6.4.8 *Project Management and Control*

Considerable emphasis must be placed on project planning. It is not possible to carry out a project of CAD/CAM integration without a comprehensive and realistic task plan. It is necessary to have a well-defined task plan at the start of a project and this plan, through the use of effective tools and methods, must be maintained and controlled as the project progresses.

One of the first tasks of the system integration Project Manager, working with a shipyard's Technical Representatives, will be to refine the preliminary task . schedules created. This wilI involve defining each significant task required in the performance of the project. These definitions will include the number of mandays each activity will require, start and stop dates, assignment of responsibility, appropriate utilization of previous system analysis, and a detailed specification of what is to be accomplished by each task.

The Project Manager will also have the responsibility for maintaining proper functional and resource utilization control of the required tasks through the use of a comprehensive in-place management control system. The interaction of these controls includes the following:

- *A Project Management System for planning, budgeting, accounting, financial control, and work progress analysis (constraint network)*
- *Project reviews and progress reporting*
- *Internal Design Review Board.*

Management control and performance visibility are strengthened through the use of a viable Project Management and Control System. A System that can schedule projects using individual resources of various types and skills, in addition to groups of resources is especially useful. However, the need to maintain a comprehensive systems concept on a technical level, no matter how abstract, is required to realistically assess how current progress will affect future aspects of CAD/CA.M system integration.

7.0 CATEGORIZATION OF SHIPBUILDING NEEDS

An upsurge in the development of CAD/CAM integration activities in industry is occurring throughout the United States, largely due to the pressures of the increasing technological potential of computer-driven systems. At present, this design/production integration effort is lacking clear-cut objectives and clearly defined functions. This lack of direction is not due to technological impediments, but to the absence of a carefully coordinated plan and unified systems approach to defining an integrated system within the environment of a specific industry. Increased awareness of the importance of CAD/CAM integration and a knowledge of areas of need within an industry, is not sufficient to enable effective integration of CAD/CAM systems. Both a knowledge of the concepts of how design-production are integrated, and a knowledge of how information, materials, and facilities interact over time are required.

Integration of CAD/CAM is based on ZOPM in shipbuilding (see Section 6.0). ZOPM is implemented via the product work breakdown structure (PWBS), the . means of planning, controlling, and reporting work efforts in shipbuilding. Thus shipyard workflow, via the PWBS, is the means. for selection and prioritizing (categorizing) shipbuilding needs for CAD/CAM integration. This thesis will be explored in this section, with a focus on the development and control of software requirements.

7.1 PRODUCT WORK BREAKDOWN STRUCTURE (PWBS)

The report on PWBS, issued starting in 1980, by the National Shipbuilding Research Program (SNAME/SPC PANEL SP-2 Outfitting & Production Aids), constitutes a milestone for the development of shipbuilding technology in the United States. A point to be emphasized is that this report provided an "...awareness of how seemingly unassociated Panel SP-2 and other (shipbuilding) projects are actively related." This emphasis highlights the use of PWBS for a pivotal role in integrated CAD/CALM system synthesis.

Using the PWBS it is possibIe to optimize efforts to reduce the redundancy of work effort and movement of materials and foster a more coherent approach to *shipbuilding work flow. Figure 7.1 depicts the PWBS as it appears in the SP-2 report in schematic form. All of the basic methods (HBCM, ZOPM, AND ZPTM) are related to a zonefproblem - area/stage orientation which facilitates integration of both actual and virtual work flow processes. These all utilize the tenets of group technology. The PPFM method enables use of Group Technology to fabricate parts, and is based on a problem - area/stage approach.*

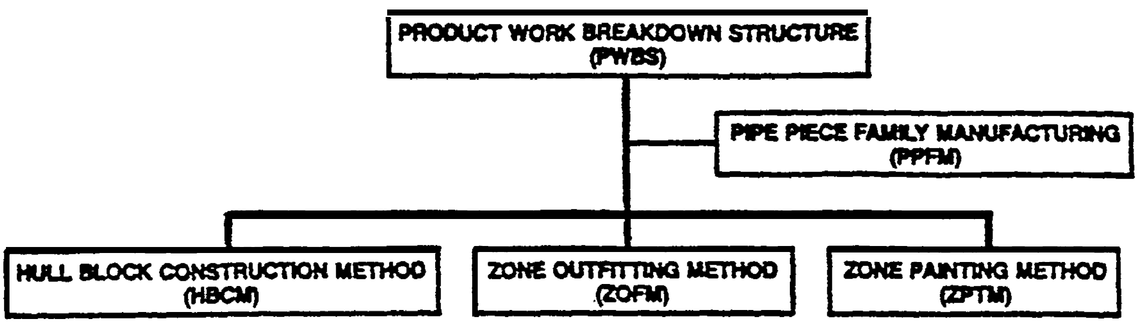


Figure 7.1: ***Schematic of Product Work Breakdown Structure (PWBS)***

A brief explanation of each facet of this diagram and important features of their application to shipbuilding work flow follows. For a comprehensive treatment of this concept see the referenced SP-2 report and related SP-2 publications.

- ***Hull Block Construction Method (HBCM)***

  Blocks must be oriented towards optimizing the outfitting process. Institution of both real and virtual flow lanes aids this process and outfitting must start before blocks are complete to enable "pre-outfitting" planning and flow. Accuracy control, applied both to planning and construction of hull blocks, can be facilitated by CAD/CAM systems. Close attention to accuracy control can prevent counter-productive work wherein completed tasks would have to be re-done when a misalignment is detected further in the work flow process.

- ***Zone Outfitting Planning Method (ZOPM)***

  Close attention to material flow planning and control in ship construction enables great productivity benefits. Design is aided by knowledge of the fabrication and assembly process, so that work instruction can be detailed and materials accurately listed. Material procurement decisions can be made faster, and lots of materials can be ordered and delivered on a by-zone basis. Use of models is made more useful on a by-zone basis and can make possible useful drawings through photogrammetric techniques.

- ***Zone Painting Method (ZPTM)***

  A knowledge of work flow and parts assembly, permitted by HBCM and ZOPM, enables coordination of painting/coating processes that would otherwise be very time consuming. Orientation of blocks close to convenient worker access can be planned to make this process faster and safer, even if done at several different time intervals.'

The above areas have been explained earlier in Section 6. Highlighted are the work flow characteristics which relate to automation of the ship design/production process.

7.1.1 PWBS and CAD/CAM Synergism

Implementation of the PWBS creates the opportunity to initiate many other productivity improvements; chief among these are Process Planning, Production Planning, Standards, and Industrial Engineering. The ability to more effectively implement these activities, integrated into the actual shipbuilding process, provides a basis from which to generate requirements for the CAD/CAM automation process. Thus, the observed synergism of CAD/CAM integration in other industries is related to the current integration philosophy of shipbuilding. Consideration of these other facets of the design/production integration process is critical to cost effective integration of CAD/CAM in shipbuilding.

7.1.2 PWBS and the Managers Function

Integration of CAD/CAM functions will be of little value if no action can be taken to effectively plan or carry out required decisions. The linking of production planning and central functions to the actual flow of ship construction via the PWBS, and applied computer systems, requires management, at alI levels, to be well versed in industrial engineering principles to enable coordination of planning, manufacturing, and assembly functions.

The absence of a cadre of middle managers who can act analytically in industrial . engineering areas has been caIled the Achillee's heel of the U.S. shipbuilding industry by Japan's foremost industrial expert, Dr. H. Shinto. Likewise, any integrated CAD/CA.M system pIanning and/or operation is doomed to failure without development of these skills at all levels of management.

Perhaps the singIemost important management focus of CAD/CAM integration is the realization that PWBS, as related to shipbuilding, is the framework of any integrated approach to design/production. Knowledge of PWBS operation, and the ability to deal with the daily decisions required to operate a shipyard (with or without computers) is the area of management training to be concentrated on for successful transition to modern methods of design/production integration. Indispensible to the success of this venture will be the presence of middle managers skilled in the tenets of software engineering to enable the use of software tools.

7.2 PWBS AND CAD/CAM

The PWBS determines the flow of work through a shipyard. The implementation of a PWBS can be done independently of investments in yard equipment, or facilities. However, PWBS effects and in turn is effected by yard facilities. Thus, there is a . key relationship between the instituting of a PWBS, yard facilities, and CAD/CAM integration. Figure 7.2 graphically shows this concept.

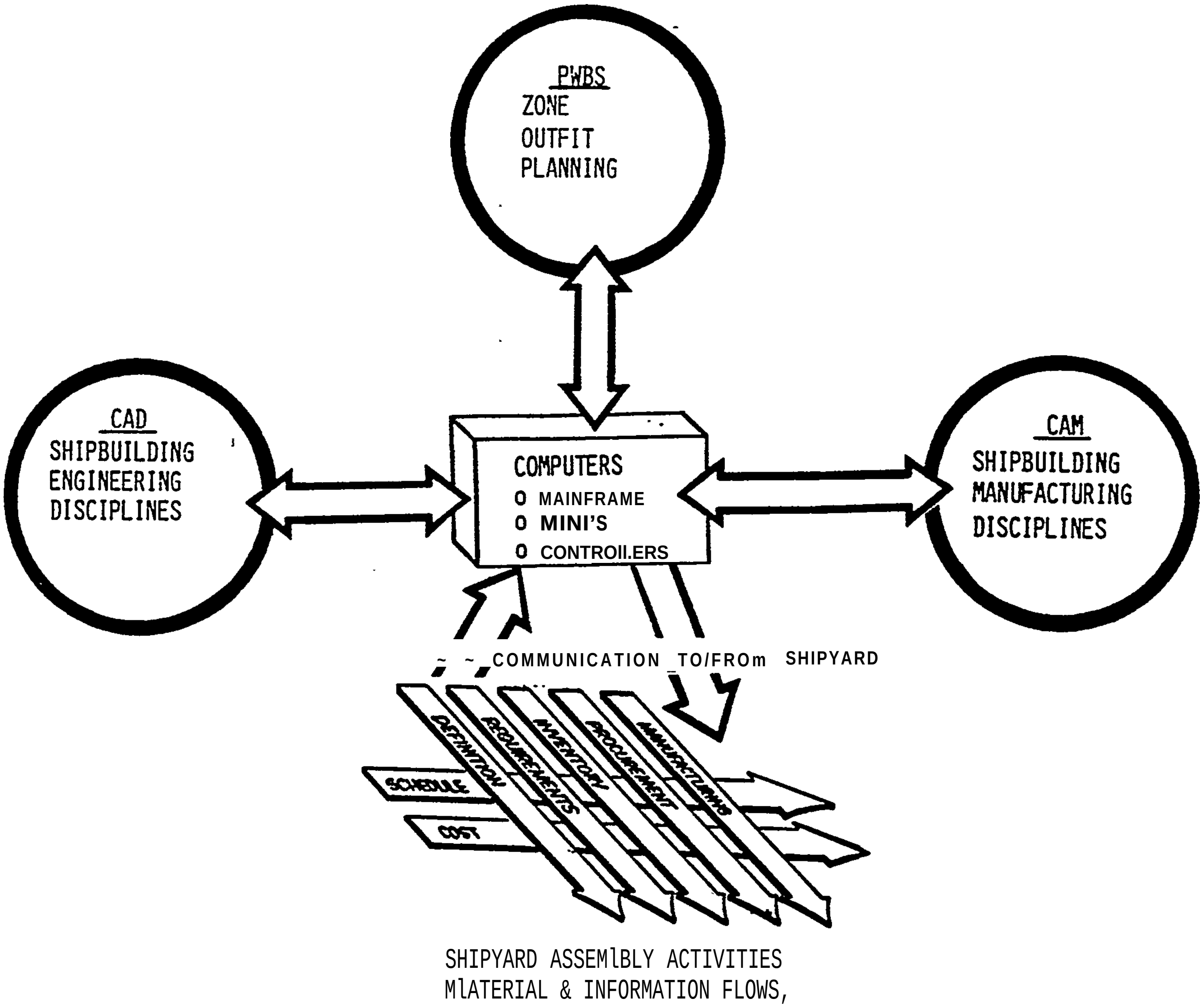


Figure 7.2 Product Work Breakdown Structure (PWBS) Interaction with Integrated CAD/CAM

This figure suggests two concepts. The PWBS should be keyed to potential computer applications, and the changing of yard facilities, caused by instituting ZOPM, should be examined for its effect on CAD/CAM. Ideally, the change in yard facilities required to make a PWBS approach work optimally should include CAD/CAM decisions. This suggests a means to implement the timing of CAD/CAM integration decisions.

7.3 SYSTEMS INTEGRATION PLAN (SIP) AND THE PWBS

Recognizing the relationship of PWBS and CAD/CAM, management can employ a Systems Integration Plan (SIP) to pace introduction of facility changes for PWBS introduction with CAD/CAM computer function changes. Special support requirements, notably software, can thus be planned accordingly. The plan for software development will thus support a meaningful CAD/CAM integration procedure and be amenable to selection of software tools to increase software productivity. Adaptation of PWBS and requisite facility modifications do not occur all at once. Numerous pIanning procedures and sub-goaIs are required to effectively implement the PWBS. These can become logical steps for software development and major goals for segments of a SIP in the CAD/CAM integration process. Figure 7.3 depicts the phased introduction of ZOPM, and the PWBS, into a yard. This time-phased plan will enable planning of achievable segments of the task of CAD/CAM integration to proceed in an orderly fashion. Importantly, it also gives logical levels of CAD/CAM integration, which once achieved, can stand by themselves and work in concert with a new level of shipyard productivity as dictated by the PWBS.

APPROXIMATE TIME - FRAME	FUNCTION/SEQUENCE BY SYSTEM	FUNCTION/SEQUENCE BY ZONE
PAST (TYPICALLY PRE- '70's)	o CONTRACT DESIGN 0 MARKETING DETAIL DESIGN O FUNCTIONAL DESIGN 0 SYSTEM ARRANGEMENTS - o MATERIAL ORDERS PRODUCTION O CONSTRUCTION 0 TEST & ACTIVATION	● (NONE)*
CURRENT (TYPICAL '84)	O CONTRACT DESIGN 0 MARKETING DETAIL DESIGN - 0 FUNCTIONAL DESIGN 0 SYSTEM ARRANGEMENTS O LONG-LEAD PURCHASE 0 SYSTEM ARRANGEMENTS 0 GENERAL PURCHASE PLANNING PRODUCTION 0 TEST & ACTIVATION	0 CHECK FOR INTERFERENCE 0 PRODUCTION PLANNING 0 SHIP PLANNING 0 CONSTRUCTION
FUTURE (LATE '80's)	O MARKETING DESIGN & PLANNING o - CONTRACT DESIGN & PLAN 0 FUNCTIONAL DESIGN & PLAN 0 LONG-LEAD PURCHASE PRODUCTION o TEST & ACTIVATION	0 TRANSITION DESIGN & PLAN 0 ZONE DESIGN % PLAN - 0 GENERAL PURCHASE 0 STAGE DESIGN % PLAN - 0 GENERAL FABRICATION 0 CONSTRUCTION

Figure 7.3-A: Time Phased Introduction of Zone Outfitting to Shipbuilding

Figure 7.3-A shows the relationship of this concept of zone outfit conversion of phases of the shipbuilding process to the PWBS and the SIP.

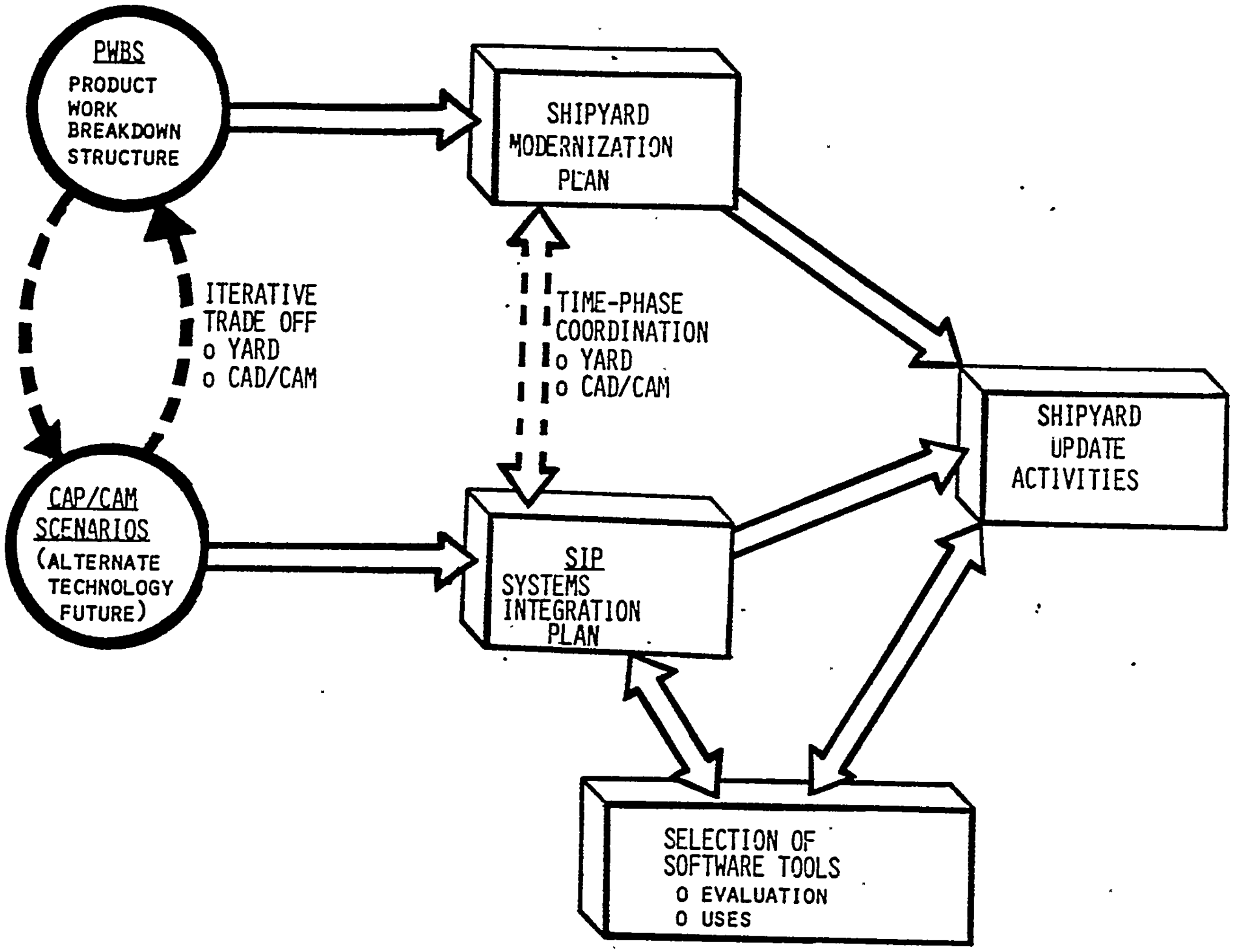


Figure 7.3-A: Relationship of PWBS to CAD/CAM Integration via the SIP

This diagram shows the means by which both the PWBS and CAD/CAM integration

7.4 *SIP AND THE ROLE OF USER INVOLVEMENT*

Shipyard CAD/CA.M integration efforts put new demands on the user. Important in any system design effort, critical to general under-the-roof manufacturing, the part the system user plays in shipbuilding CAD/CA,M integration is absolutely crucial to meaningful development of the system.

The totality of the shipbuilding environment includes design agents, customers and outside service companies as well as the inner workings of the shipyard. The need to understand ZOPM/PWBS techniques, as applied to the existing/planned limitations of yard facilities and equipment, gives a unique role of importance to the ultimate system user. The compounding of shipbuilding complexity by introduction of new ZOPM techniques make it a highly complex area, where large-block movements must be planned to optimize flexibility. Experience in the yard cannot be quickIy transferred to data processing personnel. Additionally, shipyards cannot maintain a large, full-time staff of computer analysts and programmers. Thus, it becomes expedient to concentrate on means to provision system users with the means to both participate in systems planning and aid in system review, update, and change. Knowledge to use, and access to, software tools can aid this long-term development of required functions to enable viable systems to emerge in the shipyard. Economically, this will permit continued user-system developer interface without a large permanent data processing staff. More importantly, it will provision technical requirements to create, update, and maintain a viable software development effort, via the SIP, to effect CAD/CAM integration.

Development of an integrated CAD/CAM system is a major endeavor. Selecting a system development approach where responsibilities are shared by system users, data processing personnel, and all levels of management is not the only development choice. It is, however, the only development choice that can effectively construct a worthwhile system. Many benefits are accrued if each of these three personnel areas work together well and communicate effectively. Some of these benefits and advantages of user involvement include:

- *Effective inter-group communications*
- *Professional working environment*

- *Catching of errors at a pre-programming phase - thus far less costly changes*
- *Resulting systems match actual requirements*
- *Resulting systems are <u>both</u> useable and used.*

A brief overview of what each participant in these major areas must do is as follows:

1. <u>System User Community - Users Must:</u>

- *Have assigned roles and responsibilities*
- *Have knowledge in a functional area, and this expertise must be known to others.*
- *Be made aware of the pitfulls, as well as the benefits of new technologies*
- *<u>HAVE ACCESS TO SOFTWARE TOOLS</u>*

2. <u>System Data Processing Community - Personnel Must:</u>

- *Be made aware of user role and be prepared to follow their lead*
- *Have at least a minimal introduction to shipbuilding mission*
- *Be given comprehensive review of their assigned functional area*
- *Devise technologically advanced, but not turistic, approaches*
- *Know their time lines, and have specified milestones*
- *<u>HAVE ACCESS TO SOFTWARE TOOLS</u>*

3. <u>System Management Role - Management Mush:</u>

- *Require cost/benefit analysis of solutions considered*
- *Mandate justification for selected system approach*
- *Provide support for system selection*
- *Recognize need for training and introduction of new skills*
- *<u>HAVE ACCESS TO SOFTWARE TOOLS</u>*

Software tools are the means highlighted to enable communication of system development requirements between the identified personnel groups involved in CAD/CAM integration. Importantly, this use of software tools is not only a communication link, but a repository and report generator for system development that allows building, review, and justification of a system in an orderly fashion. And, in accord with information on the software life-cycle sections of this report, the development, test and documentation of software is also facilitated later in the system development cycle. For shipyards this approach is important as major systems can be deveIoped in accordance with a SIP, while future planning can be meaningfully linked to levels of software development. The SIP itself will enable the selection, evaluation, and use of the software tools which can support the CAD/CAM integration effort with the greatest efficiency. Selection, evaluation, and use of software tools are introduced in the following sections.

8.0 SHIPYARD DESIGN/PRODUCTION INTEGRATION - A FIRST ATTEMPT

Obstacles to effective design/production integration in shipbuilding currently result primarily from lack of awareness of the existence of methods of computer-based communication, required support software, and the expense of developmental methods to plan the integration in an orderly fashion. Required disciplines of Zone " Output Planning Methodologies (ZOPM), CAD and CAM equipments are visible and known quantities and not impediments to integration. The development of integration requirements tailored for the individual shipyard must be matched with new CAD/CAM technologies. These new integrated technologies, once identified, . will enable the eliciting of the critical, yet elusive, link in CAD/CAM integration, the software issues. Thus, the integrated shipyard requirements become a first step to enable identification of the software issues.

Once identified, these software issues for CAD/CAM integration in shipbuilding can be analyzed, and plans to dimension the scope of software development, test and maintenance can be formulated. Software tools, and their applicability to enhance productivity and reduce the cost of the required software effort can then be applied on a software life cycle basis.

A hypothetical scenario depicting an "Integrated CAD/CAM shipyard of the Future" is presented in this section to enable a means of expressing requirements for CAD/CAM integration in a realistic framework. The identification of required software to effect integration, in accord with these requirements, is explained, and a rationale of how such software serves the integration function is given.

Though the integrated CAD/CAM issues discussed are real, the value of them to an actual shipyard is hypothetical, and is not meant to be construed as the best possible technology or to relate to any existing or planned system. Derivation of these integrated CAD/CAM applications are from modifications of the scenarios presented in section four of this report.

8.1 THE INTEGRATED SHIPYARD OF THE FUTURE SCENARIO

A truly "automated factory" is not a scenario goal in any industry, and certainly not the labor intensive shipbuilding industry. Perhaps the best description for an integrated shipyard would be a "paperless factory" where all data was controlled to channels where it would be capable of instant display on a graphic screen, directing the cutting of a part on an NC machine, or providing current information reports for action by yard personnel. Unrealistic only a few short years ago, this trend towards a paperless design/production capability is a very real, and achievable goal.

Internal use of computers to accomplish CAD/CAM integration will see the need for paper or mylar tapes disappearing, as the ability to rapidly down-load cutting data from a central computer to Computer Numerical Control (CNC) Devices on the shop floor become commonplace even in the most hostile environments. Data sent, via cable, would be FM modulated to enable receipt only by the desired CNC device hooked to an appropriately designated FM lModem. This expedient allows free movement of the sending, or receiving, device throughout the yard, without the need to move wires to maintain accurate receipt of data. Since each CNC device will have an affixed mini computer, with high density disc storage, actual cutting data for many parts can be stored on the shop floor and thus enable the central computer to be used for many other functions. Downloading of data to the CNC would take only a few seconds a day, yet be able to store many days of working data. This also means that if the central computer is incapacitated, no interruption in production will take place as machines on the shop floor would have available control data for days of operation. An important advantage is also afforded because the CNC computer can do post-processing of data "on-the-fly". Thus, with the use of properly formatted data as a uniform format, there will be no post processing required of any design data. This means that any machine capable of making a part, and having a CNC, can immediately create that part. This would allow, for instance, two flame cutters, in different parts of the yard to each work on half of a production order in a rush situation, or to take-over each others tasks if one cutter fails. Importantly, should a part have to be created by an outside vendor, or a follow yard, cutting data could be sent via phone line, Disc, or *magnetic/punched tape and used without additional post processing. "These described links are all examples of integration functions possible with modern computer technology. All are very cost effective, as the changes enable more production, quicker, with fewer periods of down time for tape load, rewind, etc. Quality is much better, as mis-reading of punched tape is totally avoided.*

Use of the above method also creates a very real benefit inasmuch as each CNC . can be equipped with a CRT screen to provide graphics for assembly diagrams, set-up data for a machine tool, flame cutter, or other production device. This CRT, and the mini computer it is attached to, is only used a part of the time for set-up and run instruction. This means that the ability to transmit, receive and store data at each and every CNC terminal work station is a by-product of the system. Use of an optical or magnetic wand scan system can be used to quickly "wand-in" employee identification codes and pre-defined work procedures which can be used for work in process, inventory control and work-package build data. This data can be polled from time to time, and reported back to the central computer facility automatically. Design change requests, Q.C. Data Communication, inventory, work order and safety data can also be relayed via this method. All of this gives a new dimension to MIS functions and added control of shipyard PWBS planning and execution. The above is an example of CAD/CAM synergism wherein extra benefits are accrued to other than CAD/CAM sectors of shipbuilding due to the integration of CAD/CALM functions.

All of these functions will require new software. All steps of the software life-cycle will be required for each of the mentioned applications. Not mentioned are the innumerable software programs required to control communication protocol, provide security, create screen displays, store data, prompt/cue users and many other supporting applications. The example given above, and several other examples are shown in the Figure 8.1 scenario of an integrated shipyard. A survey of software issues is provided, in abbreviated form, to outline some of the software required to operate an integrated shipyard.

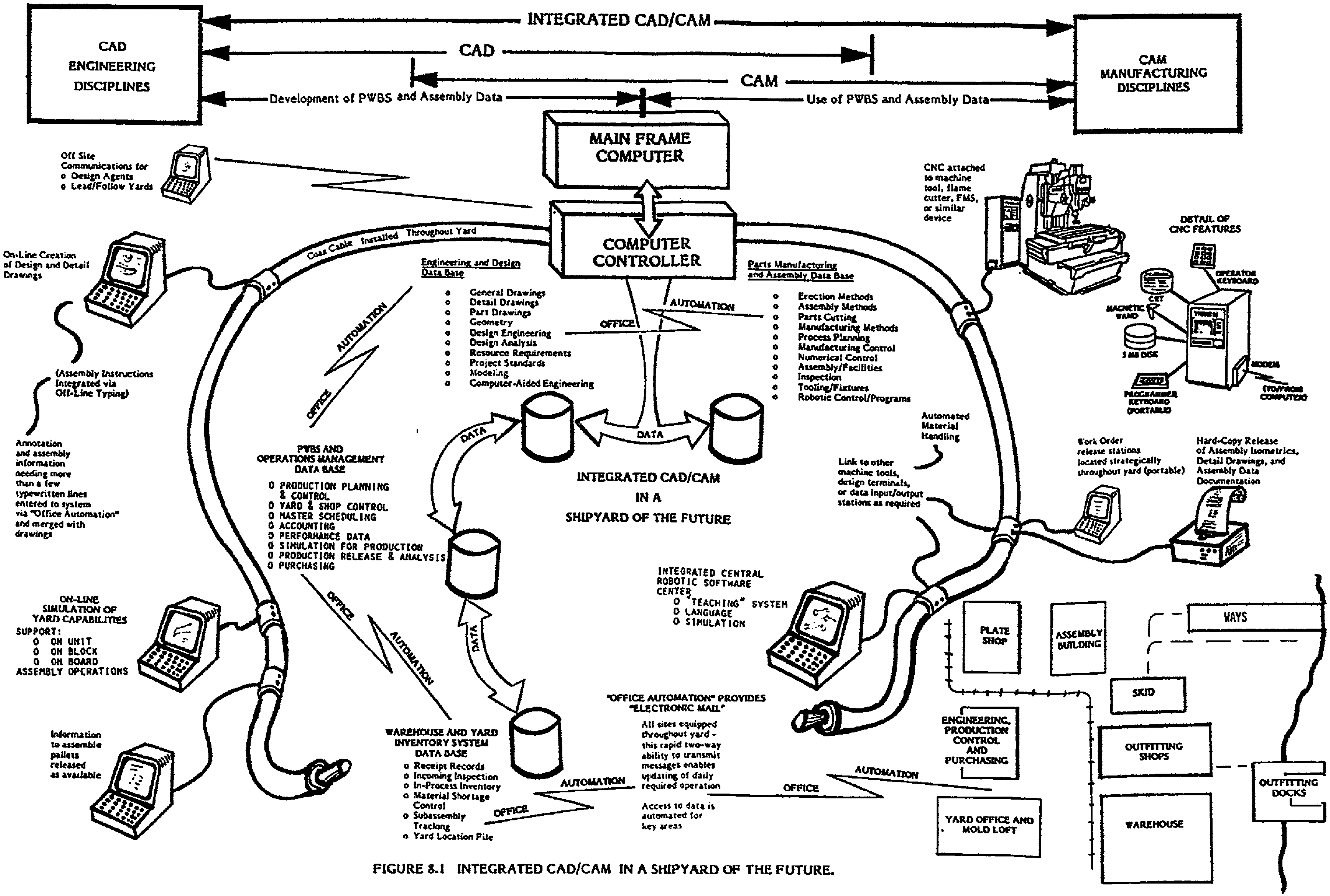


FIGURE 8.1 INTEGRATED CAD/CAM IN A SHIPYARD OF THE FUTURE.

8.2 SOFTWARE TOOL APPLICATION AREAS IN INTEGRATED SYSTEMS

Though a hypothesized scenario, and understandably a presumptuous one in many areas, the Shipyard System postulated in Section 8.1 identifies many Software Issues. A software issue is a definable system (or subsystem) requirement which will require software to enable system operation where the required software was not planned or foreseen in initial system analysis. A key characteristic of large, integrated systems is the need for early identification of software issues in order to enable resolution of these to complete the system on time and within budget.

There are three classification areas of software tasks to which resources must be applied to create an integrated CAD/CAM system for shipbuilding:

- *Specified Software System: Identified "packages" of purchased, leased, or in-house developed computer programs which are integrated into a system, by design, for a particular major function (MRP, purchasing, etc.).*

- *CAD/CAM Synergy Software: Systems, or added capability for systems, which are made technically achievable, and economically attractive by the linking of two, or more, CAD/CAM software systems. Often very unrelated to the functions of the originally integrated systems, the systems identified by this process requires a decision to create them as a part of the initiail plan, or perhaps never have the opportunity again. These systems are optional in the end result, but are important to a totally integrated system. (Instituting a shop floor data collection system when installing a CNC network for CAM is one example.)*

- *Software Issues: Identified in lead paragraph of this section. An exampIe of a software issue is the programming needed to "link" (integrate) an office automation system with a CAD system (see scenario, Figure 8.1. Software Issues arise out of decision made to integrate or otherwise modify specified, or synergy software in the CAD/CAM process, and must be resolved to make the system work.*

Once identified as part of the requirements for an integrated shipyard, these three software classifications can be isolated and grouped for review. Based on the review of the proposed integrated shipyard functions the associated specified software requirements will change. The Synergy software and software issues related to these will also change, possibly radically altering the characteristics of the system. A final decision may come down to budgetary requirements, where the projected cost-of software will be a factor. Here an iterative exercise of shipyard needs verses software beneftis offers interesting trade-offs. Elimination of a low cost "off the shelf" system may in fact eliminate a great deal of high-cost software issues required to integrate the package into the CAD/CAM system. Conversely, an apparently high-cost "in-house" system, may not engender any software issues and offer synergy choices of value to many CAD, CAM, and MIS areas. This situation may enable added budget contributions from departments who could benefit from the expanded system. Thus, the choice of systems must be closely examined prior to inclusion in the SIP (System Integration Plan). A concise, clear, statement of findings (unexpected features of an integrated system) based on the proposed scenario is a means to rigorously examine and rate the need for the classes of software required to support the integrated system.

8.3 STUDY AND SCENARIO FINDINGS

Use of a scenario appraoch to depict candidate integrated CAD/CAM systems yields many advantages. Firstly, it gives a means of highlighting findings made during the study used to formulate the system as presented. Secondly, it gives a means to enable, through analysis of the scenario by specialists from many disciplines, the "shake-out" of salient issues in the scenario, which become scenario findings. These scenario findings are clear, concise statements of problems and opportunities which are used to aid decisions on finalizing requirements for a system, and thus aid in defining the required software effort.

Use of the scenario, as a focus to effect a review of the integrated CAD/CAM system, can be accomplished by a technique called a narrative simulation. A narrative simulation requires a formal "talk-thru" of a graphically presented scenario by a person, called a scenarist, who is very familiar with the system. The *presentation is made to a group of experts in selected facets of technologies included in the system. The experts need not necessarily be familiar with the presentation, but personnel familiar with yard update plans should also be present. An extremely valuable source of participants are vendor technical specialists, who can contribute valuable insight which could be otherwise overlooked.*

After the prepared presentation is given by the scenarist, the specialists are asked to provide a "narrative simulation" of selected areas of the system and other participants are asked to contribute questions and comments. System components are configured in different combinations and relationships as determined by comments by participants. This procedure differs from a useful tool for systems analysis, the design "walk-thru", becuase it is used to synthesize system components, and not analyze a stated design.

During this narrative simulation it is important to have someone knowledgeable in software, preferably software engineering practices, present. If outside software personnel are to be used on an integration project, the narrative simulation is a critical first meeting, even if their working on the project at a later stage is not yet scheduled. Identification and trade issues for software start at this point and can effect the scope, cost, and ultimate efficacy of the integration effort. Findings made during the scenario review are termed scenario findings. These are both problems and opportunities which can significantly aid the ultimate project, and simiIarly alter both the magnitude and types of software required.

Listed beIow, with no designated importance to order of appearance, are some of the findings made during the course of this study. These study findings were made as a result of the review of the section four CAD/CALM scenarios, and were then used for creation of the Section 8 shipbuilding scenario.

- *CAD/CAM integration often emphasizes CAD-to-CAD or CAM-to-CAM "integration''... use of modern data base practices which integrates data in a central data base are more productive in the long run.*

- *Off ice automation may play a key role and offer significant savings, in-both labor and hardware in CAD/CAM integration...if carefully planned as part of the system.*

- *The manufacturing process using CAD/CAM, is often more compIex then the product being manufactured... however, the economies of having, and using these systems are worth the effort.*

- *Islands of automation (stand alone, single purpose, CAD or CAM devices) are losing their appeal...they have been recognized to be icebergs of software.*

- *Software engineering, with a life-cycle approach to all software and the use of automated software management tools, is a cost-effective must now.*

- *Organizational barriers based on traditional practices are compounded by the necessary computerization of the integration process (methods vs. design vs. MIS-etc.).*

- *Standardization will take on new meaning with CAD/CAM integration...it will become both easier to achieve and less necessary...3D CAD will enable cost-effective 5 Axis CAM machining to take place on (or off) site using software techniques; yet computer data base techniques will make selection of standardized parts, for outside purchase, much faster and easier.*

- *Standalone software to support CAD and CAM hardware will obscure the need for CAD/CAM integrated systems for several more years_... but the need to integrate will be growing all the time.*

0 Use of CAD with "three dimensional" capabilities and automation of the process of creating machine-tool. controI will make it attractive to produce more machined parts and complex sheet-metal parts in the shipyard - (this will require new machine tool investigations and investment decisions).

0 More drawings and engineering analysis wiil be done "in-house" with integrated systems... however, more drawings and analysis will also be done by design agents...these drawings will be different in type and mission.

0 Use of on-line CAD graphics capabilities to plan/simulate employment of yard facilities and resources is a valuable industrial engineering adjunct for the shipbuilding process.

0 The need for portable, quickly moved computer terminal sites throughout yard facilities argues for use of FM discrimination approaches to yard communications (coax cable/modems).

0 ZOPM for each yard is required to drive a SIP for CAD/CAM integration and allow attendant software development to take place in a logical, productive manner.

0 Software Engineerings skills, and selected software tools are a must for shipyards, as this approach can replace the need for a large, continuously in-place cadre of programmers for CAD/CAM integration at significant cost savings..

Use of these findings as a guide enables the selection and planning of required CAD/CAM component subsystems in a hypothetical shipyard scenario. In turn, the software approaches to integrate these identified CAD/CAM subsystems are derived to operate in accordance with the requirements of the developed scenario. A software tool approach can then be selected on the basis of these software decisions. Estimates of costs for planned software, and the effect selected software tools have on these costs is an important next step in determining a reallistic CAD/CAM integration plan. These issues of software cost are addressed in the next section.

9.0 ECONOMIC ANALYSIS OF SHIPYARD SOFTWARE REQUIREMENTS AND USE OF SOFTWARE TOOLS TO INCREASE PRODUCTIVITY

Many of the technological problems associated with integrating CAD/CAM functions in industry are being solved. The eventual introduction of software tools on a large scale will substantially lessen the economic impact of incorporating new technologies, thus accelerating the pace of CAD/CAM integration in industries adopting these techniques. Careful selection of software tools and consideration of the realizable economic benefits to be gained from employment of Software Engineering Techniques and tools throughout the life-cycle of Software Systems will show that the problems and initial expense associated with this introduction are far outweighed by the benefits. The economic analysis of Shipyard Software requirements and use of software tools to increase productivity is an endeavor that must be undertaken to assure realistic long-term results from the introduction of software tools to improve productivity of the CAD/CAM integration effort. The development of a corporate technique for reviewing and reaching decisions on how to develop and maintain software and justify the economics of the software tools which are increasingly a part of this equation is of paramount importance to cost control and software productivity. Software Tools and Software Engineering practices and the measurement of their benefits and costs is the subject of this section.

9.1 SOFTWARE COSTING SCOPE

Historically the appraising of the technological worth of an applications software package, and its ranking with similar packages available in the market place was the difficult part of program selection. The dollar value of each useable package in this population gave the "best cost" approach, and other things being equal, this dollar value pinpointed the "right" program to be selected, whether it be "off the shelf" or 'contract software". The impetus of productivity enhancements which are initiating a demand for highly integrated CAD/CAM systems in the shipyard is creating a need for increasingly complex software. These software systems, in order that they be properly initiated, supported and maintained, have created an entirely new environment in the shipyard, which demands new considerations when costing the software required to support CAD/CAM systems.

The general environment now emerging. is characterized by the foIlowing features:

- *Multiple computer configurations*
  - *Host and satellite computers*
  - *"Shop-Floor" computers*
  - *Business and Technical Applications Mix*

- *Extensive computer controlled communications networking*
- *Multiplicity of Software products for a single application for a single department.*
- *Interface of many departments for overlapping requirements - thus multiplicity of specifications.*
- *Software system size and system complexity are escalating for each new system.*
- *Dynamic system growth for these software systems are to become "a way of life" because:*
  - *Incremental development process*
  - *Design/manufacturing industry changes requiring CAD/CAM device updates and changes.*
  - *Addition of features to enhance operational value of system.*
  - *Addition of capabilities*
  - *System improvements*
  - *Continual correction of system deficiencies*

These environmental changes have instituted a new level of awareness of the importance of cost estimating for new systems. The scope of software costing must now be based on the life cycle cost (LCC) concept. This basis for considering software costs in their totality, not just the acquisition/coding effort, extends the decision criteria for software in many ways. These three major areas are of importance:

- *System requirements/specifications:*

  The cost of developing the criteria for each system must be considered -- even at the earliest requirement phases. Employment of resources in *these areas typically are either not tracked for every project, or are not allocated (and thus no effort expended).*

- *System test and validation:*

  Costs for effective test, both unit and system levels, must be allocated. These costs often become the "expected overrun" costs of projects.

- *Support and enhancements costs:*

  The maintaining of software and the anticipated enhancement costs are very real. These costs will vary greatly depending on selected software, means employed to acquire software (off the shelf/contract software) and projected in-house staff available to deal with software problems.

A summary of these cost allocations throughout a typical software development process is shown in Figure 9.1. After acceptance the costs of software support begin. After a short time period these support costs, especially if enhancements are required, can equate to the initial costs of software development.

In summary, costing of software must be done in a life-cycle context. When done in this manner, true systems cost comparisions can be made. This is especially true for CAD/CAM integration. Very importantly, it is in the context of life-cycle cost that the value of productivity improvements through the use of software tools can be realized. In fact, the ability to select a needed technical software approach may rest on the economies afforded through the use of a specific software tool; which tool could not have been afforded should the project requiring it not have materialized. The use of a class of software tools specifically designed to estimate software costs can be a valuable aid to this cost estimating proccess.

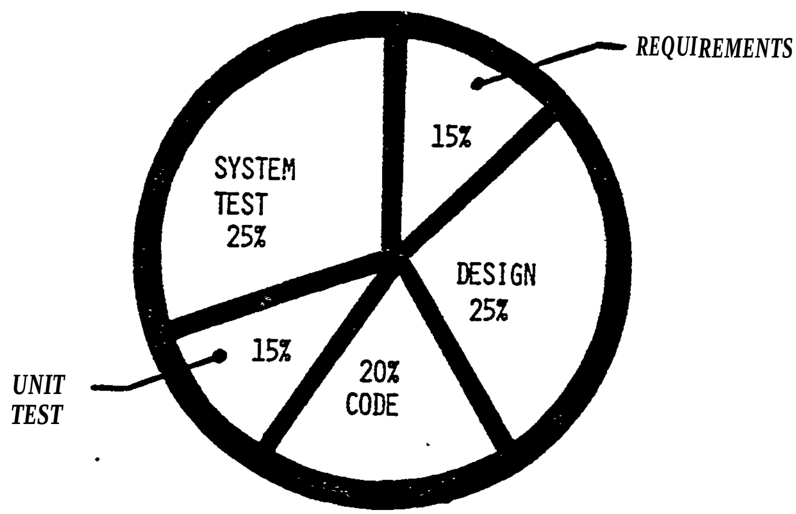


FIGURE 9.1:

Allocation of costs in a typical Software Development Cycle.

9.2 SOFTWARE COSTING PARAMETERS

The task of this study is not to investigate the current technology of software cost estimating. This would require a review of existing cost estimating approaches and assessment of their relevance and usefulness to the job of predicting CAD/CAM software costs. However, the accurate prediction of software development, operation and support costs, in the context of the Life Cycle Cost (LCC) approach is critical to determining where emphasis for software tool use should be concentrated to gain increases in productivity. Accordingly, a brief discussion of the basic precepts of software costing and predictive software cost models is described in this section.

Software is not tangible, and thus is extremely hard to describe in physically analogous terms. Thus, the first, and often most serious problem, is how to describe software parameters against which measures of progress and assignment of cost, predictive or actual, can be made. A few selected examples of software costing parameters follow:

9.2.1 Lines of Code

Though often misleading, because it does not take into account complexity of programming, individual programmer techniques or differences between languages (when more than one are being tracked), Lines of Code (LOC) is still a basic estimating parameter. Often it can be used in estimating programming and support team requirements, and also serve as a basis for estimating related requirements, such as documentation. In a given software development environment a dolIar amount per line of code, for a specified programming language, is estimated. A new project is estimated as to size, expressed in LOC, and costs calculated by multiplying LOC by the doIlar amount per line of code figure.

9.2.2 Modules

Any grouping of programming statements that can be tested is termed a module. This estimating parameter has the advantage of plotting developmental costs against defined goals via the criteria of a test, and can be approved by a Quality

Control Function, and thus enter a “change control” cycle. After initial tests, typical programming development may still require changes. The tracking of change costs on a module basis can be estimated and controlled. Likewise, the need to add “unexpected” new modules after a design freeze often comes up - these new modules can be estimated, and tracked/controlled separately. Since modules are the smallest approvable packages of software, each can be assigned to a responsible lead programmer who can act as a double-check on a modules performance and schedule.

9.2.3 Error Day

Two identical systems, performing the same tasks, but having different development histories, will have different “internals”. These unseen differences charactize their design integrity, and future error probabilities. These differences stem from the probability of error, and can be measured (to some extent) by the age of errors expressed in days. This measure is called the error day. A system developed by a top-down methodology starts testing soon after start of coding, with errors found and detected in a few days. A second system, using older development and test methods pushes off error detection until weeks, months, or even later.: This is “so because unit tests with drivers often masks errors until later .integration phases, with errors corrected in integration not interacting with other code to highlight errors until after that time. Categories of errors can be fixed at 1 day, 1 week, month and year duration. Thus 5 errors of 1 month would equal (at 20 days to the working month) 100 error days. The sum of error days from code creation to detection thus measures the quality of a system. It also is indicative of future error probalities, and the "goodness” of the design, testing and integration process. Many error days indicate high errors, attributable to poor design, whereas long--lasting errors can be attributable to poor development. The concept of error-days is not a statistic that is regularly kept, but the knowledge of this-concept allows prediction of a systems future performance, and costs, as well as ways to pinpoint areas to conserve costs in subsequent development.

These parameters are an attempt to quantify software in order that it can be priced, and some degree of predictable schedule and management control exercised. It would not be realistic to expect that this effort is easily achieved. However, it is important to recognize that the prevelant notion that the programming mystique, where all software effort is somehow magically locked in the head of each programmer, and thus not subject to any measure or control, is aIso not true. Formalized means to utilize these parameters to obtain estimates is accomplished via software cost models.

9.3 Software Cost Estimating Models

A software cost model is a systematic procedure that relates cost to certain variables or cost factors in order to estimate the costs of software development and/or software support. A computer based software model is a valuable software tool. However, the importance of understanding software cost estimating techniques in the shipbuilding environment is to enable accurate projections of software costs to better understand areas requiring the productivity increases possible through the use of software tools. Software ,managers, and importantly industry management in general, have recently become deeply sensitive to the increasing importance of software in their organizations and its rising cost. A review of software cost estimating will show that there is an enormous amount published about cost estimating models, but very few make specific proposals for sound approaches to software cost estimating on a quantitative basis. Many of these models are offered for use commercially, but need a lengthy calibration effort to make them usable in a new situation. Calibration is a process by which values of model parameters are obtained for a given cost estimating situation through use of formal curve fitting methods, representative historical data or through selecting values from experience.

Generally, the effort to calibrate a model to the software environment, or the sources of influencing forces external to the software product being developed, is very time consuming. Thus, the expense of a model’s lease/purchase must be added to the expense/time taken to calibrate it to the shipyard environment. This effort may not be worth it, or only marginally useful; however, a rudimentary understanding of how models work, and what types are available gives an insight into cost estimating techniques which is useful for software cost predictions.

Software costs are not as predictable as hardware costs for many reasons. Typically, the differences are:

- *Software development engineering is a relatively new discipline*
- *Changing cost relationships between hardware costs (trending lower) vs. software (trending higher).*
- *Software has only recently become a major cost factor.*
- *Relationship between cost and generally accepted cost factors are not established.*
- *Reliable, retrievable historical data on software is not available.*

These differences highlight the reasons why the use of software cost models output is, at best, an estimated quantity when compared with the relative certainty of hardware cost quotes.

Cost estimating methods are classified into four (4) categories:

- *Analogy (sideways)*
- *Element Estimate (bottom-up)*
- *Cost Estimating Relationship (top-down)*
- *Hybrid (combination)*

A description of how these methods work is as follows:

9.3.1 Analogy

This is a very primitive method, which compares the project under consideration with past projects, of known costs, which are similar in operation. Estimates are made by key individuals or groups of skilled personneI familiar with features of both the existing and proposed system.

9.3.2 Element Estimates

A proposed system is broken-up into component costs, and each task assigned an estimate. These estimates are sumed to obtain total resources, with schedules obtained by organizing the identified tasks by activity sequencing. This method is heavily dependent on technical and management judgment to select input parameters.

9.3.3 *Cost Estimating Relationship*

This approach, the most advanced, utilizes historical cost and resource inputs to derive relationships based on such independent variables as program size, type, memory usage, etc. The variables for the upcoming project are estimated and resources computed. Present model approaches are almost all of this type.

9.3.4 Hybrid

Some models combine the characteristics of two or more of the described methods. This allows the rapid computing of estimates to a certain level (by analogy for instance) and then a break-out into tasks for scheduling (by the element approach for instance).

9.3.5 Software Costing Model Factors

Many factors contribute to software cost. Naturally, a software estimating model must consider the cost factors of a proposed software task to be accurate. One of the chief management complaints about software cost estimates is that they are always estimated Iow, and rarely if ever over estimated. This may be because the cost factors of the proposed project, were overlooked. Similarly, if a cost estimating model doesn't take all cost factors into account, the estimate will also be wrong. In either case, the estimate will tend to be low. The important point is that cost factors in the project must be recognized, and a means to utilize these cost factors must be available in the model.

A broad categorization of software cost factors can serve as a check list for software cost estimates, as well as a means to review the completeness of software cost models being considered as a software tool. These six broad categories, and related software cost factors, are presented in Figure 9.3.5.

FIGURE 9.3.5: SOFTWARE COSTING CATEGORIES AND FACTORS

1) *Requirements Variables*
(Address System and Software)

Program type/application -
Timing requirements
On-line program
Requirements change/design stability
Response time
Security classification
Vagueness/lack of knowledge of requirements
Innovation required
Design carryover
System interface complexity

(2) *Design and Coding Variables*
(Describe site and Functions of Programs)

Number of object instructions
Program complexity
Language
Source instructions wriiten
Number of functions
Types of functions (mix)
Number of Submograms (modules)
Number object instructions not delivered
Percent object instructions reused
Percent *source instructions*
Types of instructions (mix)
Number of words in data base
Number of classes in data base
Number of input variables
Number of output variables

(3) Installation, Operation and Maintenance Variables
Impact of Support Services)

Number of user centers
Frequency of operation

4) Programming Environment Variables
(Programmer skill, Tools Available)

Programmer experience
Programmer participation
Personnel continuity
Maximum number of programmers
Percent senior analysts
Percent senior programmers
Average programmer utilization
Cost/Man
Travel required
Programming philosophy
Closed/open shop availability
Development not at operational site
Program turnaround time
Use of automated validation/ verification tools

(5) Management Environment Variables
(Impact of required responses to management)

Amount of external documentation
Schedule realism
Coupling - system/SW Engineering
Software Management Emphasis
Number of agencies concur/review
Customer inexperience
Document types required
Validation/verification responsibility

(6) Hardware Constraints
(Computer used vs programming differences)

Core capacity
Concurrent development
Number of bits/word
Machine speed
Special display equipment
Random access device
Input/output capacity

9.4 USE OF SOFTWARE MODELS

Models are used for two basic areas of software estimating: Development and Support. Development models are more. sophisticated, and focus on two classifications of system development:

- o Creation of Computer Systems that are end products, and perform a specific function. A MIS system is an example.

- o Creation of Computer systems that are an integral part of the operation of a large system. These have stringent and complex interface points with the environment. These are called "embedded" computer systems. A CNC/Machine Control Device on the shop floor is an example of this.

Software cost estimating tools (models) apply to both of these classifications. However, embedded systems must be examined in terms of the system of which it is a part, as well as the computer system itself. In either classification, a representation of the software life-cycle and its controlling environment and the design production system it is a part of, are required to specify the needs for software cost estimates. In the case of new technologies for CAD/CAM integration, the system manager must know how the software components will affect elements of cost, schedule, and risk of the systems being created. The total CAD/CAM system context must be viewed in terms of functions, speeds, reliability required and the software systems cost, schedule and risk factors required to meet these needs. Both the software manager and design/production system personnel must make preliminary cost-performance trade-offs to evaluate alternate proposed courses of technical action.

9.4.1 Estimating Precision

The use of a software model, or even a much less formal software cost estimating methodology, goes through an evolution in costing precision. In the earliest phases it may only determine if a proposed concept is totally out of reach in terms of cost or development time. This can be done by weighing one design concept against *another in large, coarsely estimated blocks of effort. Gradual refinement of design, made by technical/cost trade-off decisions, enable more detailed modeling techniques, or other estimating procedures, to yield cost estimates of ever greater precision. Final estimates are guidelines to enable decisions for commitments of funds, personnel, and scarce computer resources and demand the greatest possible precision.*

9.4.2 Output Software Cost Estimates

Many different elements comprise the make-up of software costs, and different estimating approaches utilize these elements in various ways to calculate cost parameters. A goal of most sophisticated models is to make as complete an estimate of software cost as possible, and to relate these costs directly to the software environment in which the work will be done. Typical of the estimates of elements of software development time output are the following categories:

- *ActuaI at-the-desk design, coding and testing. hours. Physical direct effort to product code.*
- *Allowances for time charged to project not directly associated with software -- this includes lost time for inefficiencies of personnel effort such as breaks, routine administration, changes, and time allocated for staff/review meetings, etc.*
- *Allowances for time not charged to project, but a part of the costs for every job. These estimates give the required overstaffing so that productive hours are included in all estimates of schedule to result in the proper net time. Time for vacations, sick-leave, training and other scheduled lost time comprise these estimates.*

These classifications give an orderly means to calculate the required time and cost figures for software tasks in the environment they must be accomplished in. Knowing the different costs for an array of these classifications enables decisions to be made on project choices.

9.4.3 Cost Estimating Situations

Whether from past projects, or subjective judgment, models use some form of prescribed input data describing system attributes. These inputs are accepted by a calculation structure that operates on the data and creates the output characterized by the model/method being used. Ideally, a model will help integrate time, effort and risk to establish a ranking of feasibility. A summary of typical software cost estimating situations is as follows:

- *Phase(s) of Development (Conceptual phase, preliminary design, full-scale development, validation, etc.)*

- *Need: Task to be estimated (analysis, component definition, monitor the progress of software systems components design development, etc.).*

- *Scope: Total Life Cycle, validation phase only, software design, module design.*

- *Level of Detail: Total system component software cost, system functional components, first level of detail for work tasks, cost per module.*

- *Inputs: System performance and function, software functions, inputs, outputs, module characteristics.*

- *Level of Precision: +/-30% through +/-10% (varies depending on need)*

- *Typical Use:*
  - *Software related costs for a CAD/CAM communication system.*
  - *Compare software costs for a 5 scope vs. 15 scope design (CAD) system.*
  - *Estimate development and costs, for a real-time, yard-wide (shop floor) data collection and CNC system to be let out for bid.*
  - *Estimated software costs for a CNC system, including training, for 3 flame and 7 machine tools.*

An important variable in these cost estimating situations is the software tools to be used to enable more productive software development at lower cost. The ability to investigate these potential savings is possible using even crude model approaches, because even if estimates are coarse, the relative affect of using software tools can be investigated prior to project start.

9.5 SOFTWARE TOOLS AND SOFTWARE COSTS

Software tools affect both the rate and quality of software production. Their use extends from the system identification phase of software requirements through the maintenance and update of systems in active use. Realizing the mechanisms and parameters for the effective creation, tracking and control of software are different for each phase of the life-cycle, software tools must be matched to the phase or phases of the software life cycle being considered. The effectiveness of software tooIs used for a project is more dependent on the interaction and interdependence of the tools selected, then on any one tools features. ‘Thus, there are many trade-offs for tools selection, with a tools leased/purchase cost being a major parameter. Since, the mission of any tool is to increase productivity, the data collected for software cost models can be used, Lines of Code (LOC) for example, to rate the effectiveness of software tools. In turn, the tangential benefits of increasing LOC produced per day may affect the software environment, for example, by actually reducing the number of support personnel required. This will change the inputs to the software cost estimating model being used. Using this approach, the effect of software tools on software production rates, personnel requirements and system costs can be examined. Very importantly, the value of software tools must be examined in the total context of the support required for all shipyard CAD/CAM update activities. This can be done by checking each candidate software tool against potential specified software, synergy software and linking software efforts identified in the system integration plan, or deduced from potential software efforts elicited from the CAD/CAM scenarios. This exercise yields a cost basis for evaluation, selection and potential uses of software tools (refer to Figure *7.3-A for a flowchart schematic of this process). More than likely, the benefit of using a software tool cannot be cost justified for any one effort. However, placed in a larger software engineering context, the selection and use of* *tools, carefully evaluated for cost-benefits on several projected projects, will reap major benefits. The key to successful implementation of a set of software tools is a realistic appraisal of anticipated costs without tools (usually higher than anticipated), and a true picture of the potentiaI benefits which can be accrued with a properly implemented software tool usage plan for all life-cycle phases, and all anticipated CAD/CAM software requirements.*

9.5.1 Positive Software Tool Cost Factors

A number of positive factors are accrued through use of software tools, which have a bearing on system cost. "The factors are apart from the productivity increases in lines of code, and any or all may apply given the situation and task at hand. A judgmental. decision on which of these factors apply, and a cost estimate of the value of the seIected factor can be applied to each of the items listed:

- *Automates status update of documentation and controls documentation costs.*
- *Enables rapid and reliable incorporation of changes.*
- *Increases management, technical and contractual visibility and control.*
- *Facilitates test and integration cycles.*
- *Augments Quality control process.*
- *Gives history of test discrepancies and "Fixes".*
- *Promotes image of competence to contract monitors.*
- *Gives visibility into program structure.*
- *Traceability of requirements through design to product structure.*
- *Integrated ensemble of programmer tools and aides for software production, and test.*
- *Facilitates software transition to new generations of computers.*
- *Greatly lessens learning curve of new software programmers -eliminates disasters of experienced software personnel leaving.*
- *Reduces labor costs*
- *Gives historical data for future cost control.*
- *Automatic enforcement of standards.*
- *Management overview of software status, availability.*

Many of these supportive features of software tools may not be relevant to currently in-place small software operations in shipyards today. However, increasing need to install and utilize software will make consideration of these features important.

9.5.2 Negative Software Tool Cost Factors

The development of recommendations for installation of software tools to support the shipyard CAD/CAM software effort would not be complete without consideration of the negative cost aspects. These Costs are both directly and indirectly expressible in dollars. These factors, aside from the actual cost of the software tools, are:

- *Some increased front-end costs at project start-up not traditionally a part of normal software projects.*
- *Unfamiliarity on the part of most, software/engineering and project directors/managers with software management leads to a reluctance to institute formal controls/changes in procedures.*
- *Need to "tailor" software tools selected ,to each yard's environment required because "off-the-shelf" solution does not exist for entire software life cycle.*
- *Personnel concerns related to increaed management visibility, peer exposure, utilization and data collection discipline must be dealt with at onset.*
- *Need to have available computer resources to operate tools must be planned/maintained.*

These software tool problems are for the most part, one time only cost/problem areas. Noteworthy is the comparison with positive features, which are virtualIy all recurring cost problems, which software tools minimize on a continuing basis once in use.

9.6 SOFTWARE TOOLS AND SOFTWARE PRODUCTIVITY

Cost of software tools for systems development can be analyzed by three different approaches.

- *Cost-effectiveness evaluation: Analysis of costs and performance without an analysis of economic benefits.*

- *Benefit analysis: Measurement of expected benefits with no consideration of associated costs.*

- *Cost-benefit analysis: A means of analysis which atempts to discern if the stream of benefits resulting from the use of a defined set of software tools, within a given level of performance, is greater than the required investment. It also can determine whether a stated level of investiment is optimal for a maximizing of net benefits. This method is quite different from the previous two in that it can provide sufficient data for making an investment decision.*

Benefits from the use of software tools arise from the performing of tasks faster and qutantitatively better than previous methods, with no lessening (or even an increase) in quality or work output. Accrued benefits are measured relative to previous methods of accomplishing tasks. The problem is to assign dollar values to the benefits "measured. Benefits can be measured by either empirical test (benchmarks) or parametric analysis.

An empirical test requires two control groups, and a specified (typical) programming task. One group does the task using the existing or baseline method, the second group then performs the same task using the software tool or tools being investigated. Benefits measurable when comparing the use of tools vs. no tool use conditions are:

- *Cost savings accrued through tool(s) use to attain the same level of performance as the baseline method.*
- *Benefits from performance levels accrued from use of tool(s) over and above the baseline method.*

Use of this approach, even with a very small task, can usually be arranged with the cooperation of a software supplier. Rough order of magnitude savings in the test situation can be extrapolated to the appropriate software tasks estimated by methods explained earlier.

Parametric analysis is performed by decomposing tasks into many base elements and then assessing the benefits from improved performance. This allows a comparison of performance differences on each element using the baseline method and software tools approach. Once these comparisons on elements of tasks are completed, each application being investigated can be reconstructed through reconstituting software tasks from the elements and knowledge of the life-cycle environment appropriate. The proportion of each element in the tasks are specified by the parameters, and benefits from doing a particular application with software tools can be estimated.

9.7 COST BENEFIT ESTIMATION

Cost-benefit analysis of software tool use is a combination of cost-effectiveness and benefit analysis comparisons. Cost effectiveness analysis identifies approaches with the best performance at a stated investment level and/or least costly approaches for a given level of productivity. However, selection of an approach from a number of different software-tool usage mixes at differing levels of cost and performance is not addressed, since no way to judge if added improvements in performance are worth their added costs is provided. A benefit analysis places a value on added performance levels, and thus addresses this point. A cost-benefit analysis systematically compares benefits from several approaches with their assigned costs to highIight the one with the greater net benefit. Net benefit is simply gross benefit minus total cost. Optimal price-performance functions are *obtained by performing cost-effectiveness analysis for a variety of approaches at various cost and performance levels. This results in an optimal price-performance function, obtained by:*

- *Iteratively calculating the best Ievel of performance at a selected level of investment.*
- *or, calculating at different performance levels the approach to reach this level at least cost.*

The result is termed an "efficient frontier", which depicts approaches optimized at all levels for cost as a function of performance, or performance as a function of cost.

A corresponding "Efficient Frontier of Benefits" gives the maximum benefits level possible for any cost level. The resulting data given is calculated from benefits corresponding to data points on the previously calculated efficient price-performance frontier. Maximum benefits possible for each level of performance is the resulting output of this formulation.

The cost-benefit analysis attempts to locate the software development approach having the most net benefits, or maximum benefits minus costs.

A plot of this function is presented in Figure 9.7. Note that the benefit function must exceed the cost function to make any particular approach worthy of consideration. Note also that the area of net benefit has a performance range having a maximum and a minimum value. Since software cost/sizing estimates are often fraught with uncertainty, it is important that this range be a wide one. The cost function shows decreasing marginal efficiency of capital, because as performance is improved, an increasingly greater cost is associated with each increment of performance improvement. Similarly, the benefit function shows diminishing returns because additional increases in performance yield smaller and smaIler increases in benefits.

The. three analysis methods presented enable a guide to investigate the effect of software tool use on software systems yet to be started. Though promoted as tool for investment decisions, cost-benefit analysis is too gross and difficult to use at the system level Since methods to accomplish benefit anaIysis are not well developed, cost-effectiveness analysis will remain as the easily used method to choose between alternatives when selecting software tools. The sizing of these software systems, and ability to estimate costs, gives a planning guide to CAD/CAM system costing. Investigating the costs and benefits associated with decreasing the costs of the proposed software systems through use of software tools enabIes valuable additional cost savings to be a realistic resource in the pre-planning process for CAD/CAM integration.

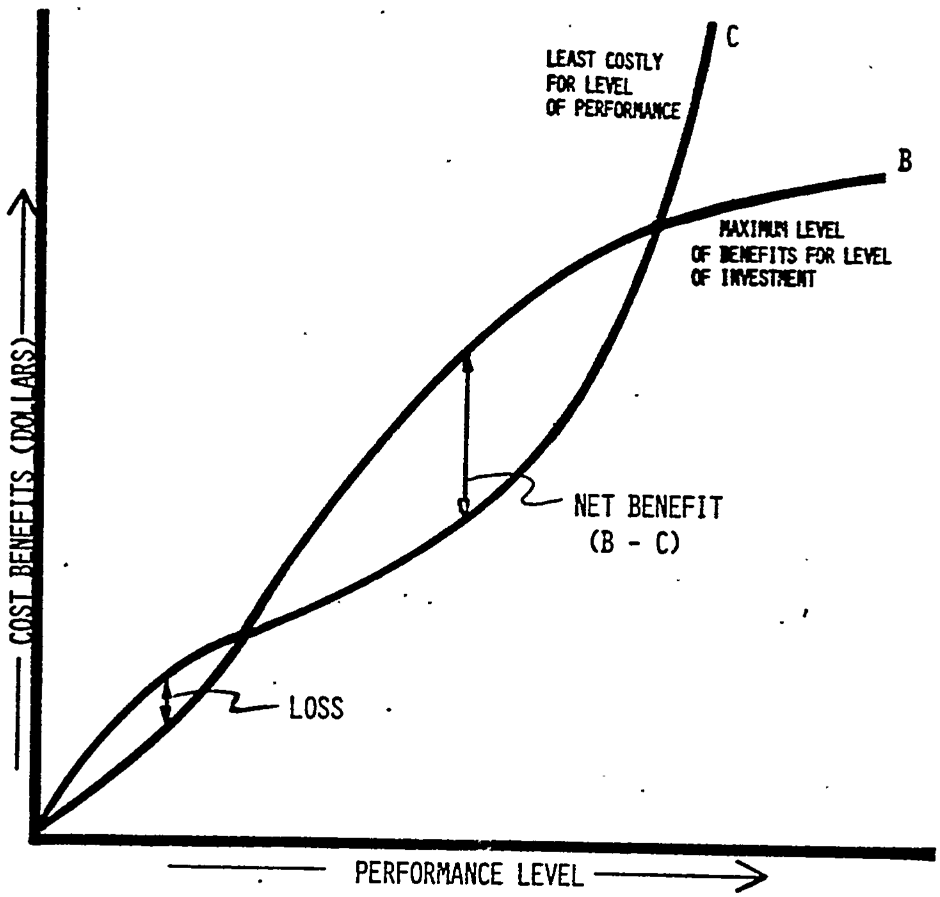


FIGURE 9.7 COST BENEFIT CURVE

10.0 CATALOG AND CLASSIFICATION OF SOFTWARE TOOLS

Software Tools are a rapidly developing technology area, thus any catalog and classification of the topic area will necessarily be incomplete. To minimize the effects of this situation, the major classification source document referenced in this section is the National Bureau of Standard (NBS) Software Development Tools special report which provides an excellent baseline of both software tool categories and specific software tools. Utilizing this report as a baseline enables, through future reference to the Institute for Computer Sciences and Technology (ICST-part of NBS) Publications an update to the data presented. Through the provisions of the Brooks Act the ICST has a mission to develop standards for . . . " economic and efficient purchase, lease, maintenance, operation and utilization of automatic data processing equipment by Federal Departments and agencies". A part of this effort involves studying and evaluating methods that enhance the productivity and quality of software. One method to attain better quality software at increased productivity rates is to utilize computer technology itself to aid the process. Thus, automation of the software development/procurement function itself becomes an aid to development of automated systems, such as Integrated CAD/CAM for Shipbuilding. The NBS quotes from a GAO report which states that the use of software tools can provide the following benefits:

> *Better management control of computer software development, operation, maintenance, and conversion.*
>
> *Lower costs for computer software development, operation, maintenance, and conversion.*
>
> *Feasible means of inspecting both contractor-developed and in-house-developed computer software for such quality indications as conformance to standards and thoroughness of testing.*

The ICST developed their software tool report, classification schedule, and data base of software tools based on the GAO report and its findings. The growing number of complex software tools, their applications and the means of evaluating them can be aided by these precepts. More importantly the means to gain an insight into further development in software tools can be obtained by knowldege of the NBS source, which also includes listings of software tools in the public domain.

The following data is largely developed from this NBS report. An addition of information on a certain classes of tools excluded from the NBS report, International activities in the tools area, and inclusion of broader terms in the original classification schema has been made to make the data more applicable to Integrated CAD/CAM for Shipbuilding.

Importantly, mention of any commercial product implies neither endorsement nor recommendation by the authors of this report, NBS, or any sponsoring agency of this report. Mention of specific product names is done only to specify typical - tool availability, and is not meant to represent the best, or only tool available.

10.1 A PERSPECTIVE ON SOFTWARE TOOLS

Tools and the knowledge to use them is the theme of this report. Context of use is extremely important when selecting a software tool, or class of software tool. The task for which a software tool ultimately will aid creation of a computer system is important, but the mating of software tool user capabilities is equally important. The perspective taken has been the life-cycle of software, and a multi-level view of software tool classes. Figure 10.1 shows a simplified software life-cycle and types of software tools applicable.

A view of each of the segments of the software life-cycle will show that there are specific software tools which can be used to support each area or areas, Figure 1O.1-A depicts a series of these possibilities, using "typical" real world ,software tools. The use of the software life cycle depicted in this figure, or the assignment of tools to each stage may not apply to a particular software environment. This caution of using both a suitable life cycle description, and tools assigned to the specified environment is a key to making software tools workable in a given corporate structure.

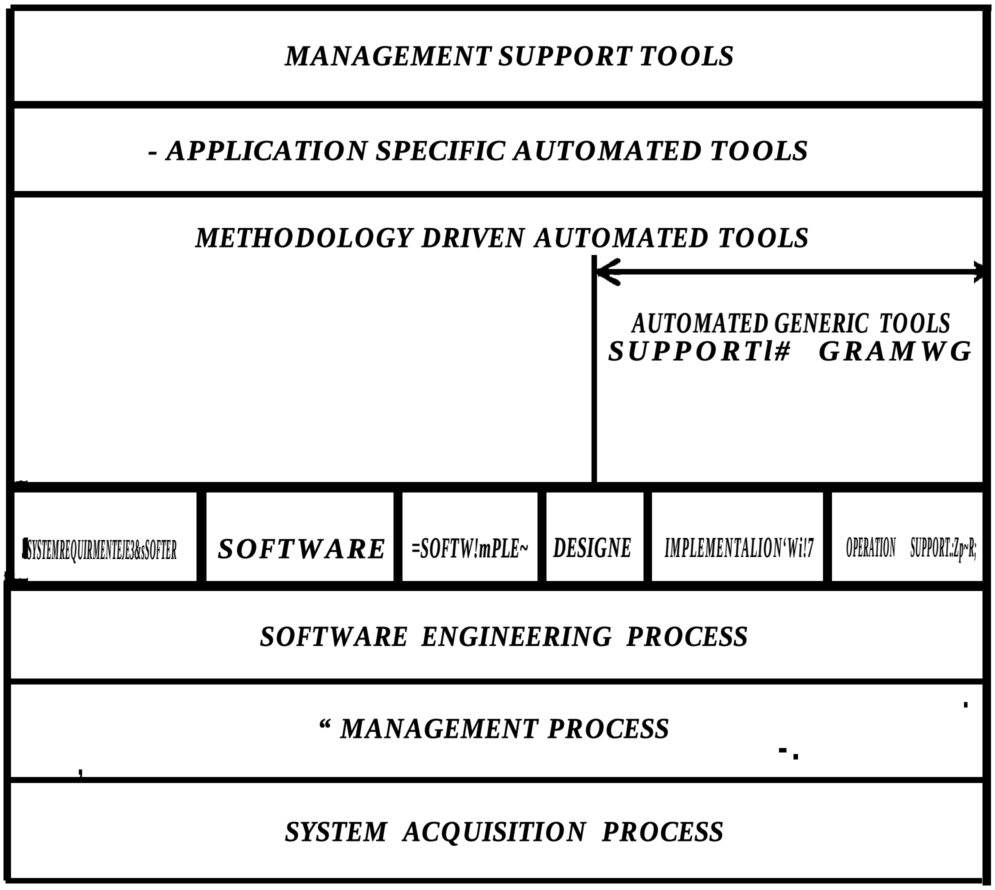


FIGURE 10.1: SOFTWARE, LIFE-CYCLE AND CLASSES OF SOFTWARE TOOLS

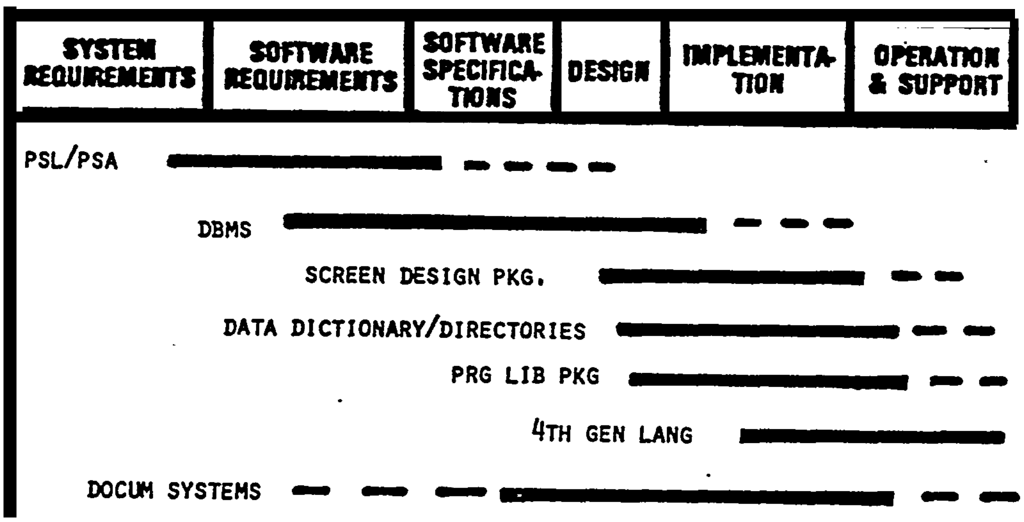


FIGURE 10.1-A: LIFE-CYCLE PHASES WITH SPECIFIC SOFTWARE TOOLS ASSIGNED

There are many ways of classifying software tools at a more precise level than the life-cycle associations presented. However, the need to be certain that all life-cycle phases are covered by use of a software tool, or tools, with minimaI unnecessary redundancy, will go far towards ensuring the full use of the power of software tools.

Example of software tools from a data oriented view would include the following types of tools:

- *Data Base Management Systems*
- *Data Dictionary/Directory*
- *Report Writers*
- *Data Base Design Aids*
- *Application Programming Languages*

Example of software tools for a user oriented perspective would include:

- *Program Development/Programmer Productivity Aids*
- *Automated Documentation Systems*
- *Language-to-Language Translators*
- *4th Generation Languages*
- *eHi Level Languages*
- *Program/Project Management*
- *Word Processing Networks*
- *Screen Descreen Formatters*
- *Editor - Syntax Development Languages*

Other types of software tools include:

- *Security Management Packages*
- *Access Control Programs*
- *Micro Processor Utilization*

These software tools are utilized throughout the software life-cycle, and support the continum of development in major projects, such as CAD/CAM integration. A graphic example of software tools as a foundation for a major software porject over time is presented in Figure 10.1-B. A very important rationale for instituting *software tool use, as shown in thisFigure, is that they support system growth, especially in the technically uncertain areas, such as CAD/CAM integration.*

The difference between software synthesis and the software development process is shown in Figure 10-1-C. The phases of the software development process are shown, together with some of the many variations by which some of these phases may be named, or sub-divided into, in Figure 1O.1-D. Within each of these developmental s, there are generic functions, tasks of various types, and related documentation requirements. These components of each phase are outlined in Figure 10. lE. The task and functions, as outlined in Figure 10.I-E, are used as a basis to assign software tools. An example of the assignment of functions and tasks to each phase is shown for the design phase (Figure 1O.1-F), Development Phase (Figure 10.I-G), Integration Phase (Figure 10. 1-H), and, Development Phase (Figuree 10.1-1). Knowing the functions of each phase in the life cycle indig~nous to a shipyard is one important means of locating and assigning potential software tools.

The discussion of software tools that follows departs from this key issue of application of software tools to life cycle phases, and discusses software tool attributes by function. Functions of tools, a means to classify them, hardware and software characteristics, availability, and sources of tools, are topics also touched on to enable the location and use of tools in the software life cycle.

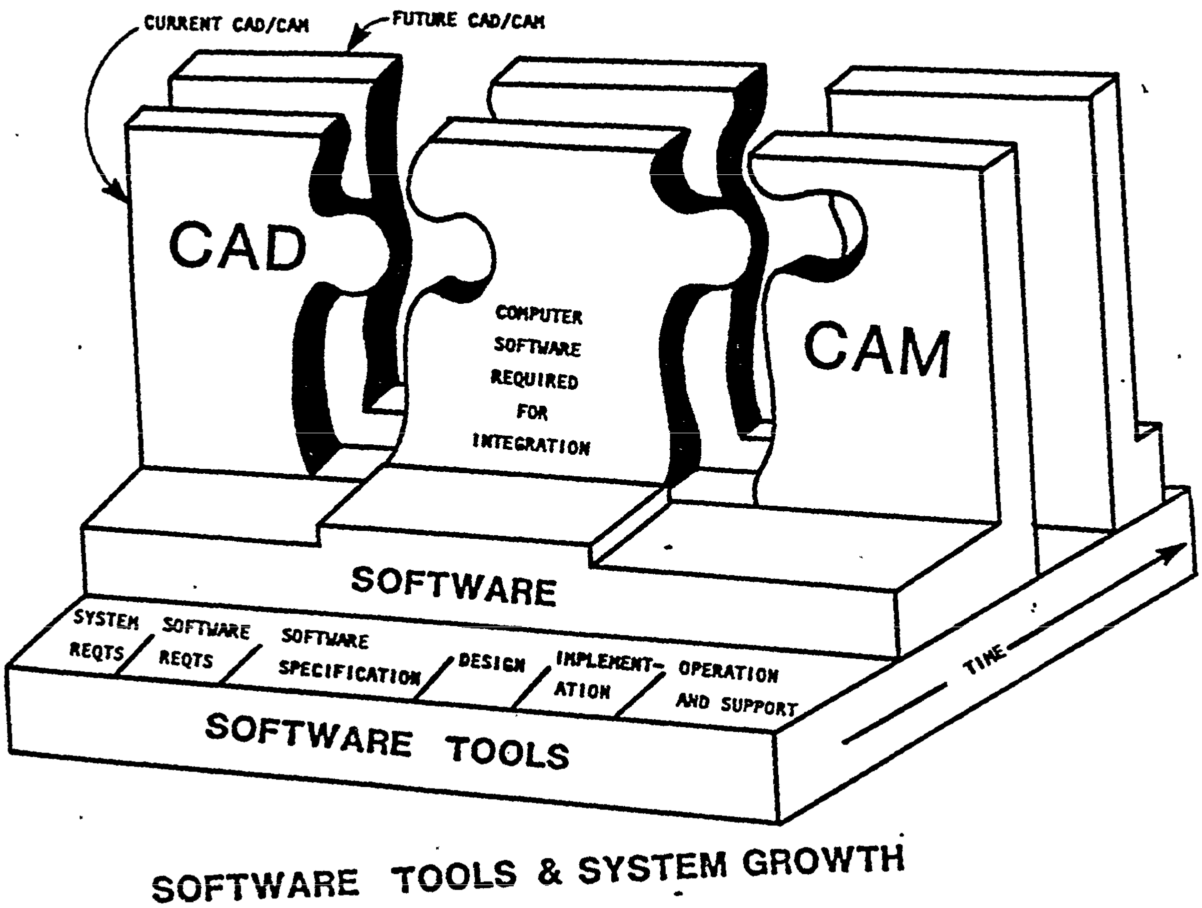


FIGURE 1O.1-B: Role of Software Tools as a Foundation for CAD/CAM lntegration

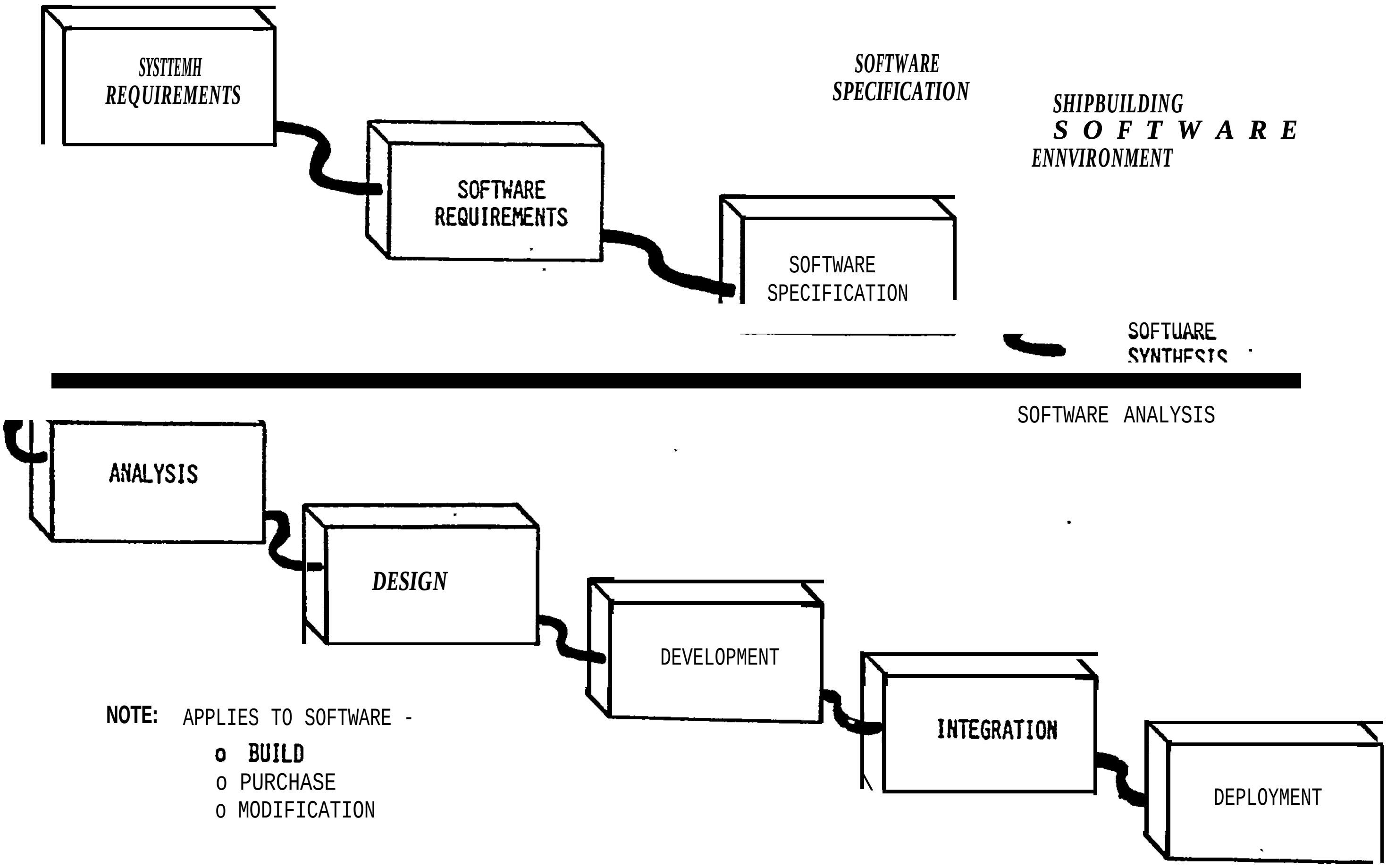


FIGURE 10.1-C: Software Synthesis and Software Development

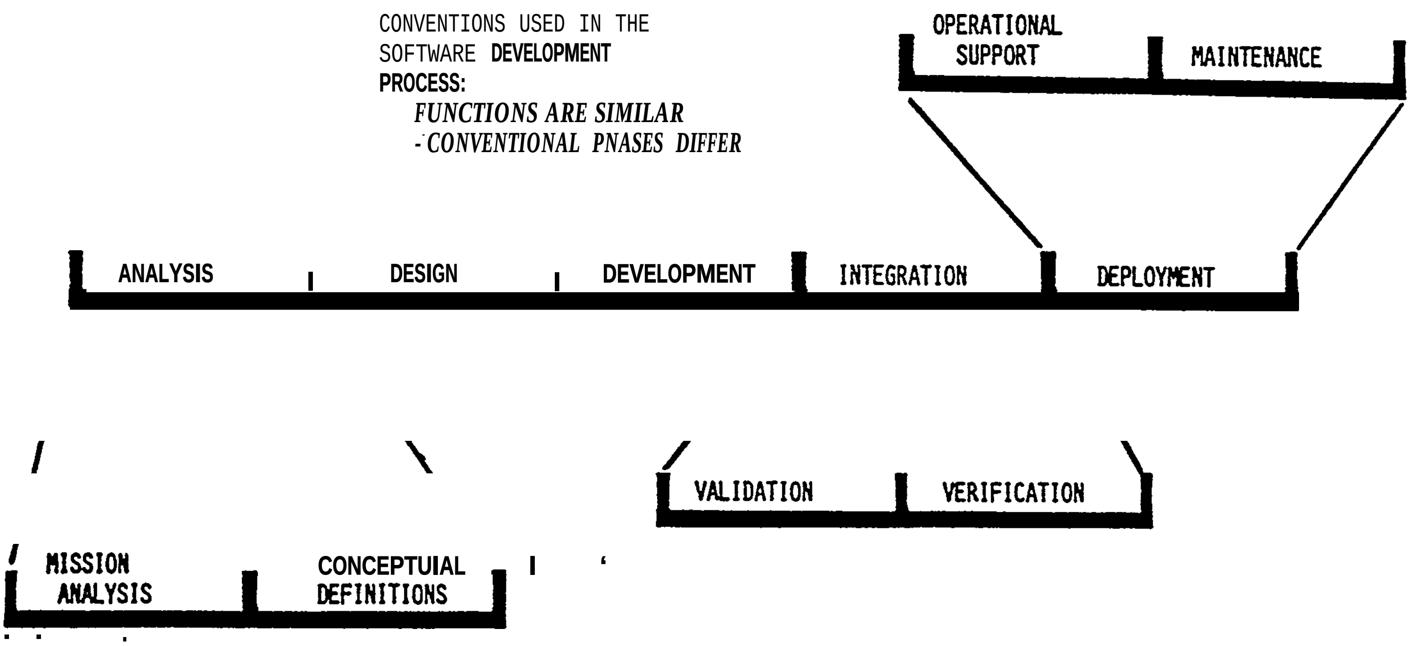


FIGURE 1O-1-D Software Development Process Phases

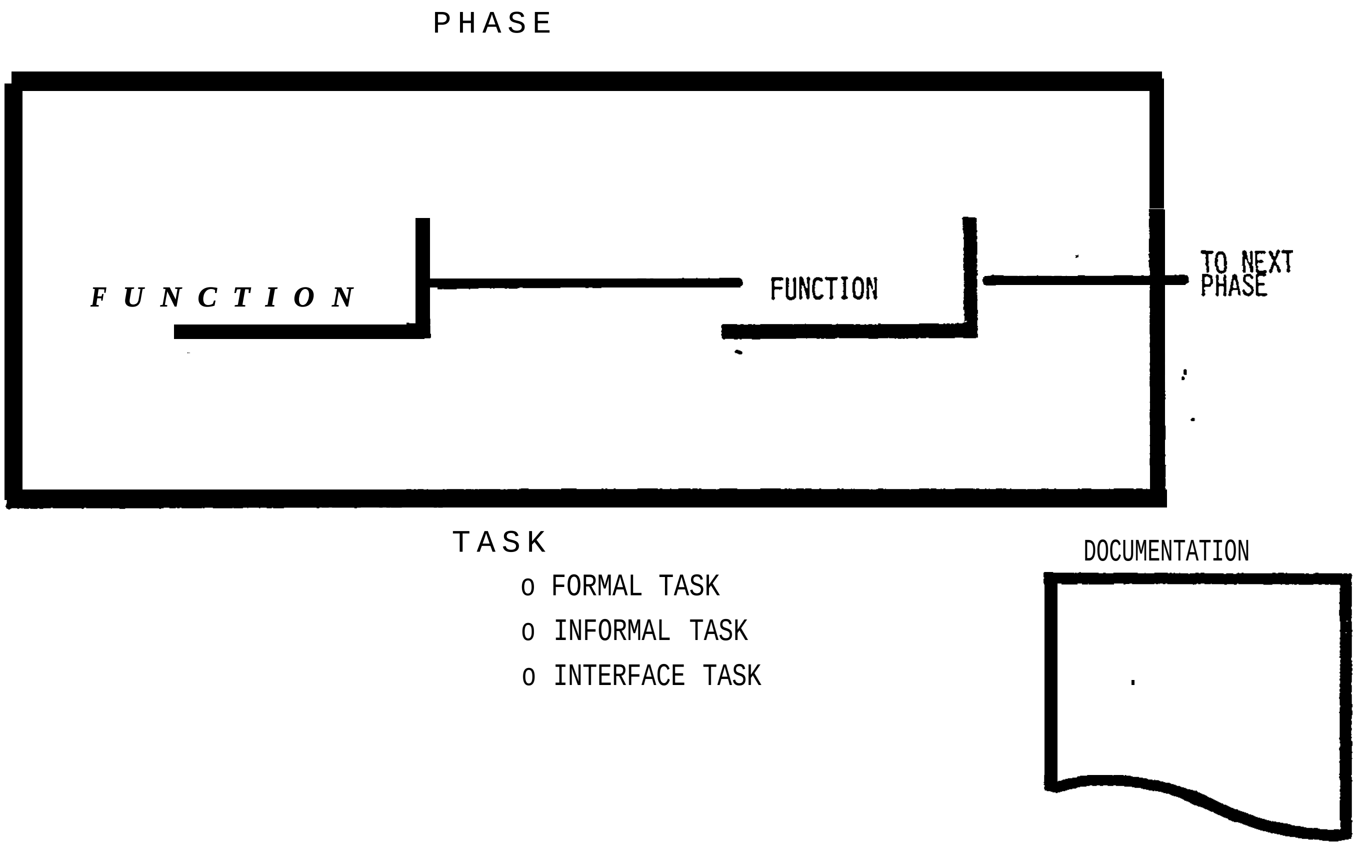
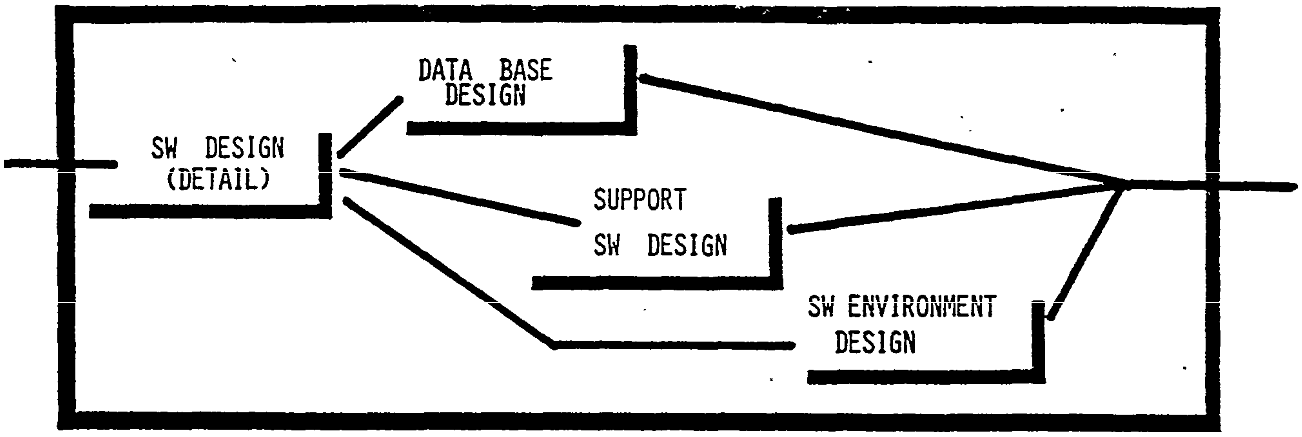


FIGURE 10.1- & Software Tasks and Functions in Development Phases

DESIGN



- o As to existing Spec. and Plan
- o Component Design
- o Modeling
- o Storage
- o Timing
- o Scheduling
- o Control

- o *Data Base Schema/sub-schema*
- o *Data Base Strwcture*
- o *scrccn Design*
- o *Data control*
- o *Security*
- o *specialsoftware Software Requirements*
- o ***operations***

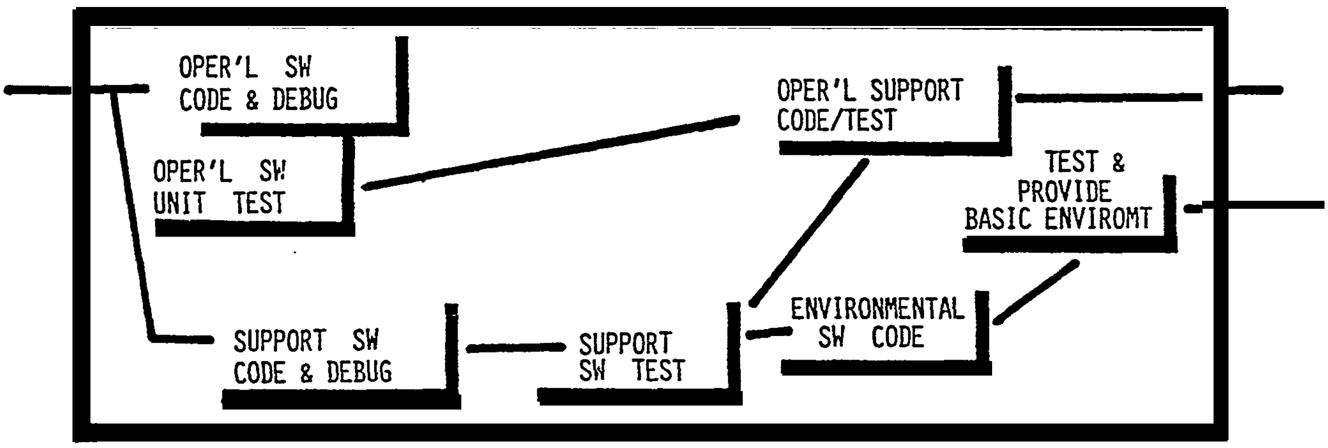
Test Datak Plan

- o Software Designs
- o Specify Operating System Design
- o Standard Environment

Subsystem Specifications
Program Specifications
Interface Documentation
Test Plan Draft
Configuration Control Documentation

FIGURE 10.1- F:Tasks and Functions of Software Phases- Design,

DEVELOPMENT

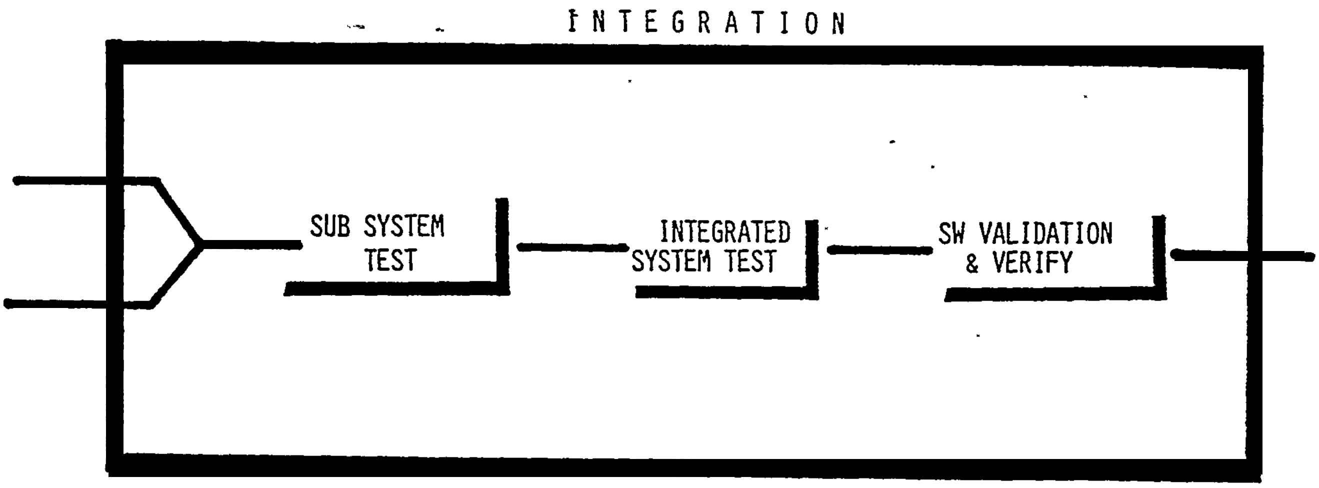


- o Program/Module Code
- o Executed and Debug Code (Unit Test)
- o Test Performance
- o Revised System Flow
- o Stand-Alone Test Data
- o Preliminary Data Reduction

- o *Test Plans*
- o *Training Plans*
- o *Screalandouq)m Fcumatting*
- o *Evahation of Readliness*

- o Data Base Manual
- o Program Manual
- o Unit Test Data
- o Integration Plan

FIGURE 10.1-G: *Tasks and Functions of Software Phases - Development "*

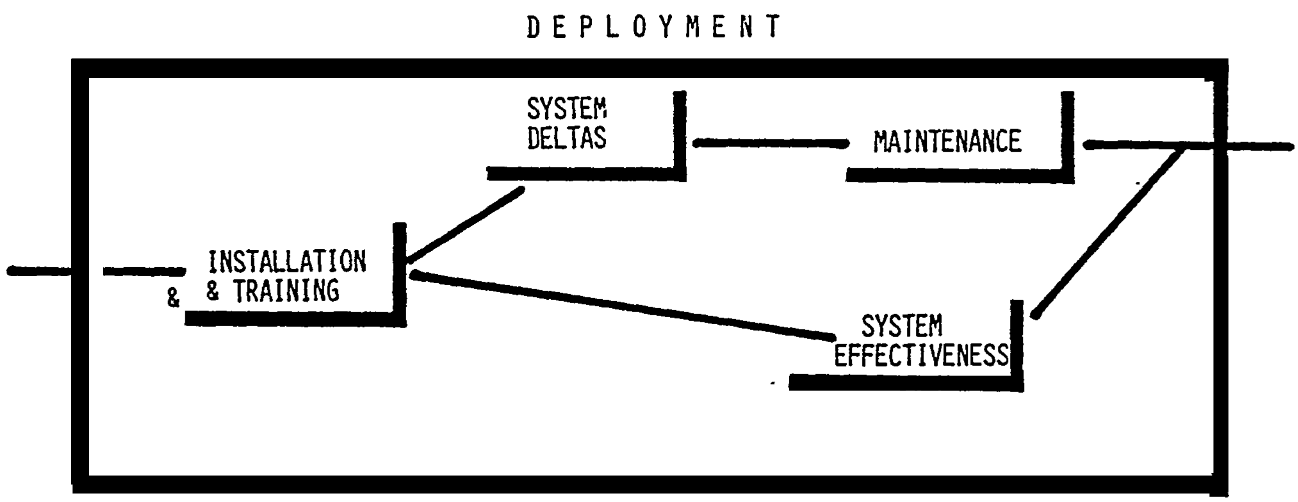


- Discrepancy Tracking of close-out
- Test Conduction and Doct.
- Finalize Performance Testing

- Tuning
- Software changes to Integrate
- End-to-End Tests
- Degraded Mode Testing
- Reliability - Maintainability Interface
- User Involvement

- Operations Manual
- Final Users Manual
- Test Results
- User Acceptance
- System Performance Data/Reports
- Integration Results

FIGURE 10.1-H: Tasks and Functions of Software Phases - Integration



- User Training
- Installation and Test
- System Start-up

- Benefits Analysis
- Cost Comparisons
- Effectiveness studies
- Management Surveys
- Performance Analysis

- Tunning and Investigations
- Changes to Baseline
- Unique Installation Requirements

Complete System Doct's.

Change Processing and Release Doct's.

Maintenance Doct's.

FIGURE 10.1-I: Tasks and Functions of Software Phases - Deployment

10.2 SOFTWARE TOOL CLASSIFICATIONS BY FUNCTION

Coincident with the growth of computer system sophistication has been the increasing complexity of software tools to aid in the development of systems and application software to operate on these systems. Starting in the 1975 era software tools emerged, and the "Buzz Words" in" vogue were compilers, debuggers, dump analyzers, flow charts and editors. Progress has evolved an entirely new series of terms, such as software development systems, application generators, soft ware engineering facilities, progam generators, and programing environments. A classification schema comprised of six divisions was put forth by the NBS to categorize these tools. These classifications of software tools are:

- *o Software Management, Control, and Maintenance Tools (MAC) - 33%*
- *o Software Modeling and Simulation Tools (SAM) - 14%*
- *o Requirements/Design Speeif ication and Analysis Tools (RAD) - 14%*
- *o Program Construction and Generation Tools (GEN) - 10%*
- *o source Program Analysis and Testing Tools (TAA) - 34%*
- *o Software Support System/Programming Environment Tools (ENV) - 3%*

The percent figure follows each in the frequency of occurrence in the NBS survey of 360 Software Tools. Added to the above are the following specialized classifications:

- *o Data Directing Systems (DDS)*
- *o Generalized Data Base Management System (GDBMS)*
- *o Software Cost Estimating System (SCES)*
- *o Foreign Tool Activity*

As the classifications presented are not mutually exclusive, and do not aIl apply to a limited portion of the software development cycle, these must be considered only generally descriptive terms. Tools in each category will be briefly explained and actual examples given in the software tool catalog section of this portion of the report.

10.3 SOFTWARE TOOL CLASSIFICATION BY FEATURE

A convenient way to provide a broadly applicable classification of software tools is to classify features of software tools. These can be applied to the functions, and give a more concise description of each individual tool. Figure 10.3 shows the classification of software tools by feature.

The following sections, 10.3.1 through 10.3.3, define each of the terms in the software tools features diagram. This figure may be used as an index to section 10 information by use of the noted paragraph numbers.

10.3.1 Software Tools: *Input Classifications*

The input features to a tool fall into two categories:

One is the control input (how the tool should operate), the second "is the subject input (what the tool should operate on). Each of these is expIained below:

10.3.1.1 SUBJECT IN PUT- *usually the main input to a tool which is operated on by the main functions -the tool performs. The following are the major types of input for tools.*

CODE INPUT

Accepts a program written in a high level, assembly, or object level language. Code is the language form in which most programming solutions are expressed, and tools are, in most cases, further classified according to the specific language that they accept.

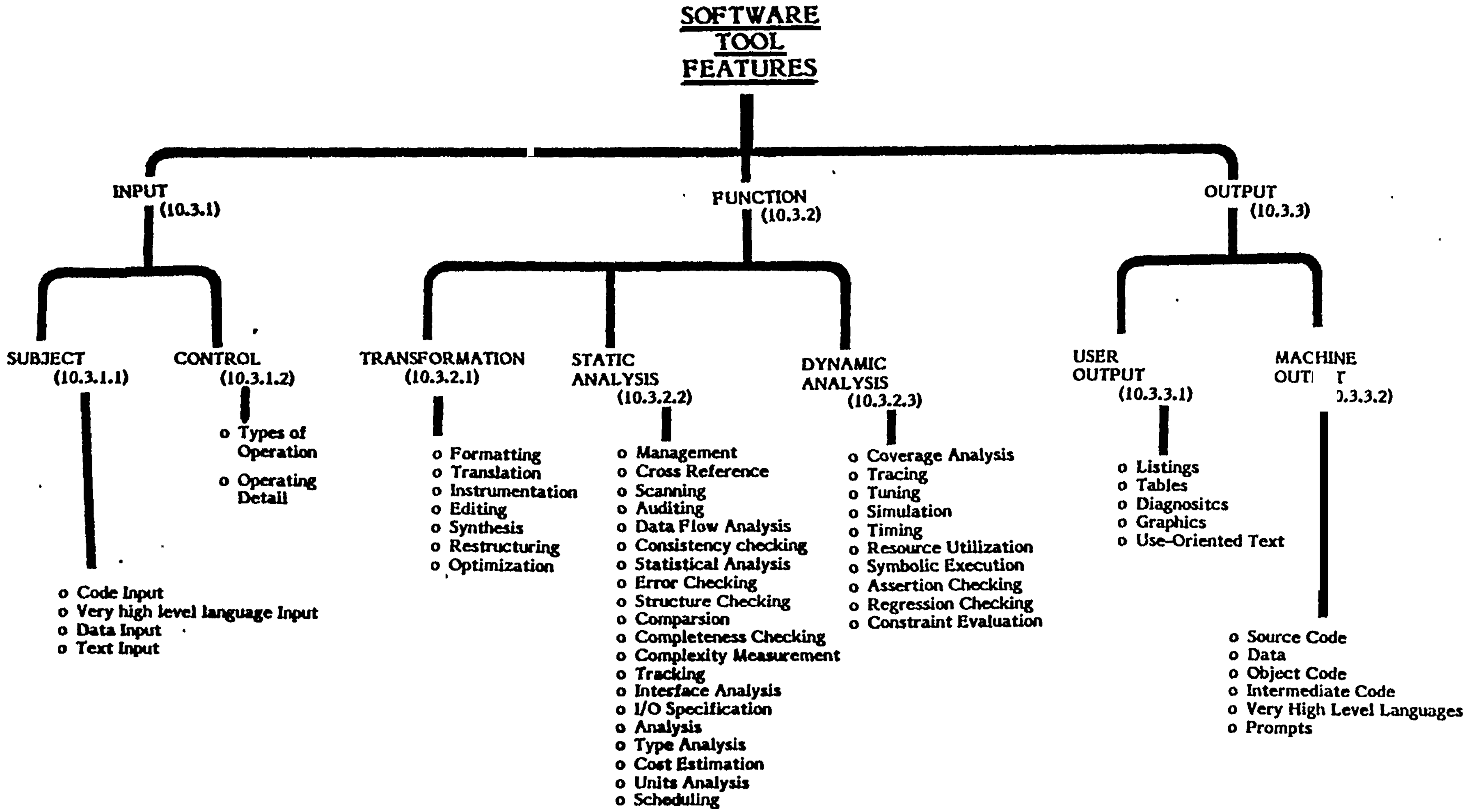


FIGURE 10.3,: Classification of Software Tools by Features

VERY HIGH LEVEL LANGUAGE (VHLL) INPUT

Accepts a program written in a very high level language that is typically not an executable form. Tools with this feature may define programs, track program requirements throughout their development, or synthesize programs through use of some non-procedural VHLL. These include tools for Design Specifications, Requirements Specifications, Program Specification, Requirements Language Aids, Design Languages, System Specification and lModel Specification.

DATA INPUT

Accepts a string of characters to which meaning is or might be assigned. This input, or raw data, is not always in an easily interpreted, natural language forin.

TEXT INPUT

Accepts statements in natural language form. This includes text editors, document preparation systems, and requires no other input except directives or commands. .These tools are oriented toward the development of documentation, and because emphasis of the database is on tools that are specified to software development, there are many tools that have text as input that are not included in the data base itself.

10.3.l.2Control Input

Tools that have control input features accept statements or data. The type of operations and any operating details associated with the tool use are in this category of description. Features in this area are difficult to determine from tool descriptions. This does not imply that control input features are not important. They relate to the user interface, wtich in many cases determines user acceptance or rejection of a tool. Tool descriptions generally lack this type of information, even though it is a principle feature that determines whether or not a tool is actually utilized by personnel.

10.3.2 Software Tools: Function Classifications

Processing functions done by a tool are included in this class. The three sub-divisions are Transformation, Static Analysis and Dynamic Analysis.

10.3.2.1 Transformation

These features describe how a subject is manipulated to accommodate user needs. They describe what transformations take place as the input to the tool is processed. Operations in this area, and related data, are as follows:

- ***FORMATTING***

  Arranging a program according to predefined or user defined conventions. These tools clean up program variable declarations, indenting statements, and making other standardizing changes.

- ***TRANSLATION***

  Converting from one language form to another. Includes compilation, structure preprocessing, macro expansion and conversion. Few situations enable these tools to cover 100 percent of the translation process, so allowance for a degree of manual inspection/translation must always be made.

- ***INSTRUMENTATION***

  Adding sensors and counters to a program for the purpose of collecting information useful for dynamic analysis. Most code analyzers instru-- ment the source code at strategic points in the program to collect execution statistics required for coverage analysis and tuning.

- ***EDITING***

  Modifying the content of the subject by inserting, deleting, or moving characters, numbers, or data.

- ***SYNTHESIS***

  Generating an application or program from a specification or from an intermediate language. Tools that have this feature include application generators, program generators, compiler compilers, and preprocessor generators. Importantly, tools with this feature show particular promise toward increasing programmer productivity; thus there is considerable emphasis on new development in this area.

- ***RESTRUCTURING***

  Reconstructing and arranging the subject in a new form according '-defined rules. Generation of structured code from unstructured code is an exampIe of the function of this classification of software tools.

- ***OPTIMIZATION***

  Modifying a program to improve performance, to make it run faster or to make it use fewer computer resources while accomplishing the same function is the mission of this classification of software tools. Many yendors' compilers provide this feature, but there are also many tools that claim this feature, but do not really modify the subject program. Instead, these tools provide data on the results of execution which may be used for tuning purposes to enable optimization of software code.

10.3.2.2 Static Analysis

Static analysis features describe operations on the subject without regard to the executability of the subject. Described in this section is the manner in which the subject is analyzed. There are many feature headings in this classification, those defined below being the primary ones:

- ***MANAGEMENT*** - *Tools that aid the management or control of software development. Since this is such a broad area, only the topic headings of tools in this category are listed as an introduction to tool features covered.*

  Configuration Management
  Global Variable Management
  Project Management
  Data Base Management
  Change Control
  Test Data lManagement
  Files Management
  Library Management
  Version Control
  Documentation Management
  Performance Management
  Capacity Planning
  Management Planning

Management tools aid in creation of .an environment to get software work done. Due to this feature, these aids typically deal with the support of the entire life cycle to aid productivity through facilitating the software itself. Accordingly, it is often difficult to tailor these tools to the management process, unless the philosophy of software tool use becomes a part of the managerial practice itself.

- ***CROSS REFERENCE***

  Tools that reference entities to other entities by logical means.

- ***SCANNING***

  Tools that examine an entity sequentially to identify key areas or structure.

o ***AUDITING***

Tools that conduct an examination to determine whether or not predefined rules have been followed.

o *DATA FLOW ANALYSIS* .

Tools which perform graphical analysis of the sequential patterns of definitions and reference of data.

o ***CONSISTENCY CHECKING***

Tools capable of determining whether or not each entity is internally consistent and that it contains uniform notation and terminology consistent with its specification.

o ***STATISTICAL ANALYSIS***

Tools performing statistical data collection and analysis.

o ***ERROR CHECKING***

Tools that can determine discrepancies, rank their importance, and highlight probable cause.

o ***STRUCTURE CHECKING***

Tools that detect structural flaws with ih a program (improper loop nestings, unreferenced labels, unreachable statements, or statements with no successors).

o ***COMPARISON***

Tools *determining and assessing differences between two or more items.*

o ***COMPLETENESS CHECKING***

Tools assessing whether or not an entity has all its parts present and if its parts are fully developed.

- ***COMPLEXITY MEASUREMENT***

  Tools determining how complicated an entity, routine, program, system, etc., is by evaluating some number of associated characteristics. Complexity factors include instruction mix, data references, structure-/control flow, number of interactions/interactions, size, and number of computations.

- ***TRACKING***

  Tools that track the development of an entity throughout the software life cycle. These tools check the software development/resource use on a module level, the module intefaces, and test progress/discrepancies. Thus, they become a vital aid to Quality Control, Management, and Documentation Development.

- ***INTERFACE ANALYSIS***

  Tools that check the interfaces between program eIements for consistency and adherence to predefined rules. Highlighting of analmous conditions is a valuable function of these systems.

- ***1/0 SPECIFICATION ANALYSIS***

  Tools analyzing the input and output specification in a program, usually for generating input data.

- ***TYPE ANALYSIS***

  Tools that evaluate whether or not the domain of values attributed to an entity are properly and consistently defined.

- ***COST ESTIMATION***

  Tools which assess the behavior of the variables which impact life cycle cost. These can range from manual methods to sophisticated computer models. Section 9 of this report deals with these features in detail.

o ***UNITS ANALYSIS***

Tools determining whether the units or physical dimensions attributed to an entity are properly defined and consistently used.

o ***SCHEDULING***

Tools used for assessing the selected software development schedule and its impact on the software life cycle.

10.3.2.3 Dynamic Analysis

Dynamic analysis features specify operations that are determined during or after execution takes place. These features differ from static features because they require some form of symbolic or machine execution. They describe the techniques used by the software tool to derive meaningful information about a program's execution behavior. The list that follows outlines dynamic analysis features. Coverage analysis, training and tuning tools are the most commonly used.

o ***COVERAGE ANALYSIS***

Tools determining and assessing measures associated with the invocation of program structural eIements to determine the adequacy of a test run. Coverage analysis is useful when the user is attempting to execute each statement, branch, path, or iterative structure in a program.

o ***TRACING***

Tools tracing the historical record of execution of a program. Because of its broad coverage, tracking has been further extended into the categories of path flow tracing, breakpoint control, logic flow tracing, and data flow tracing.

0 ***TUNING***

Tools determining what parts of a program are being executed most often.

- ***SIMULATION***

  Tools representing certain features of the behavior of a. physical or abstract system by means of operations performed by a computer.

- ***TIMING***

  Tools reporting actual CPU times associated with the running of a program or its parts.

- ***RESOURCE UTILIZATION***

  Tools providing an analysis of resource utilization associated with system hardware or software.

- ***SYMBOLIC EXECUTION***

  TooIs reconstructing logic and computations along a program path by executing the path with symbolic, rather than actual values of the data.

- ***ASSERTION CHECKING***

  Tools enabling checking of user-embedded statements that assert relationships between elements of a program. Checking may be performed with symbolic or run-time data.

- ***REGRESSION TESTING***

  Tools which use the rerunning of test cases which a program has previously executed correctly in order to detect errors caused by changes or corrections made during software development and maintenance.

- ***CONSTRAINT EVALUATION***

  Tools that accomplish the generating and/or solve path input or output constraints for determining test input or for proving programs correct.

10.3.3 Software Tools: Output Classifications

Features of software tools that provide links from the tool to a human user, and/or to a test (or target) computer are termed Output Features. These Features describe types and forms of output produced by a software tool.

10.3.3.1 user output

The features describe types of information returned from a software tool to a human user and the forms in which these outputs are presented. These user output features include:

- ***LISTINGS***
  Output that lists source programs and/or data.

- ***TABLES***
  Output arranged in parallel columns to exhibit a set of facts or relations in definite, compact and comprehensive form.

- ***DIAGNOSITCS***
  Output that indicates what software discrepancies have occurred.

- ***GRAPHICS***
  Graphical presentation with symbols indicating operations, flow, etc. Graphics features are also categorized in the following areas:

  Flow Charts
  Hierarchical Trees
  Design Charts
  Activity Diagrams
  Charts
  Hipo Charts
  Line Graphs
  Bar Charts

Control Maps
Histograms
Milestone Charts
Activity Diagrams
Structure Charts

- ***USER-ORIENTED TEXT***

  Output that is in natural language form. These include documentation and reports.

10.3.3.2 Machine Output

These features handle the interface from tool to a non-human user. Output can be directed to a target machine or to another software tool for additional processing. These features describe what the receiving tool or machine expects as output. The list that follows shows the tools that fall in this classification.

- ***SOURCE CODE***

  A program written in a procedural language that must be input to a translation process before execution can take place.

- ***DATA***

  A set of representations of characters or numeric quantities to which meaning has been assigned.

- ***OBJECT CODE***

  A program expressed in machine language which is normally an output of a given translation process.

- ***INTERMEDIATE CODE***

  Code that is between source code and machine code.

- ***VHLL***

  A program written in a Very High Level Language (VHLL).

- ***PROMPTS***

  A series of procedural operators that are used to interactively inform the system in which a software tool operates that it is ready for the next input.

10.4 SOFTWARE TOOLS AND REQUIRED ENVIRONMENT

The environment required by a software tool depends on the degree to which the tool is portable. A software tool is portable if:

- *It is written in a portable subset of a language*
- *A federal standard was adhered to in the language it is written in*
- *Three, or more, different manfuactures' Computers of different architecture can be used to operate the to'ol.*

A software tool is classified as "partly portable" if available on different computer manufacturers of different architecture, or minor modifications are all that is needed to operate on different machines.

The language a software tool is written in is a critical environmental factor. A tool must be able to be accepted by the repertoire of languages that are supported by the facility in which it is to be used. FORTRAN and COBOL are the most commonly used languages to support software tools.

By far the most important environmental factor is the hardware requirements to support a software tool. Some require a specified manufacturer, while others may be operated on a given hardware system as an accident of development and their use on other machines only awaits testing. IBM, CDC, UNIVAC, HONEYWELL, DEC and AMDAHL are common Hardware " Systems that support many software tools. Care must be exercised in selecting a Hardware environment for any given software tool, as the need for a specified software operating system may also be a requirement. Use of an entirely separate, sometimes radically different hardware system to . , support software tools, software development, or both is a possibility.

10.5 SOFTWARE TOOL AVAILABILITY

Software tools are developed by many types of organizations for many reasons. Many software tools are developed for private use by a company where only information about the tool will be shared, but the tools themselves wilI not be released for sale to the public. However, a surprisingly large number of software tools are available to the general public at no, or only minimal, cost. These tools are in the public domain, and represent a rich source of information about tools and software tools for use by interested organizations. Sources of software tools in the public domain include:

- o National Technical Information Service (NTIS) Computer Products Support Group.
- o Federal Software Exchange
- o Computer Management and Information Center (COMIC)

Developers of Software Tools include ,commercial reserach organizations, commercial vendors of software tools, universities, government agencies, U.S. supported research centers, foreign governments and even individuals. The growing interest in software tools is making many new sources of information available, but a basic understanding of software tools, and many very good software tool systems, are available in the listed public domain sources.

10.6 CATALOG OF SOFTWARE TOOLS

Following are a list of Software Tool descriptions, and their availability. Many of these are in the public domain, others are available from commercial institutions.

JAVS - JOVIAL AUTOMATED VERIFICATION SYSTEM

TOOL SUMMARY: JAVS is a program which supports a methodology for systematically and comprehensively testing computer software. The *methodology uses the structure of the software undergoing test as the basis for analysis for Automated Verification System (AVS). JAVS itself is effective for both individual and cumulative software test cases. A capability to facilitate the construction of test data that will thoroughly exercise the software, and an analysis of retesting requirements following software modification is also included. JA.VS can provide the following:*

(1) Analyze and format source text

(2) Perform flow analysis

(3) Insert instrumentation for performance measurement

(4) Describe inter-module relationships

(5) Generate test measurement results

DC2, DATA CATALOGUE 2

TARGET PROCESSORS: IBM, Honeywell, Univac

RESTRICTIONS: Marketable product

TOOL SUMMARY: Data Catalogue 2 is an independent Data Dictionary-/Directory System. It supports many data base management systems, and can interface with many (source) languaes. DC2 provides ability to enter and querey or report upon data, relationships, procedures, the raw materials if data processing, and non-computerized information such as forms, documentation, users, resources and procedures. Features include a sophisticated security system, variety of user initiated options, complete documentation, and wide base of user acceptance.

Contact: TSI International, 187 Danbury Road, Wilton, CT-06897

MULTI-LEVEL EXPRESSION DESIGN LANGUAGE - TEXT PROCESSING

RESTRICTIONS *(copyrights, licenses, etc.); Contact Martin Marietta for detalis, P.O. Box 179, Denver, CO-80201*

TOOL SUMMARY: MEDL-X will provide the user with the ability to interactively assemble, edit, analyze, and publish the software documents which, when combined, form an integral component of the development process. MEDL-X will not be merely another text editor 'or word processor installed on a minicomputers. The usefulness and power of MEDL-X are derived from its ability to employ the information contained within the files associated with the other MEDSYS Processors in addition to its own database. Additionally, MEDL-X allows the text of standard B PLATE") paragraphs to be stored within a "BOILERPLATE" file for subsequent inclusion into a given document. The format and content of a given software document is determined by the standards of the customer or the software developer. By allowing the "rules" associated with a given document to be stored within an easily updated, MEDL-X will maintain its ability to serve, irrespective of the volatility which may affect a given set of standards.

SREM - SOFTWARE REQUIREMENTS ENGINEERING METHODOLOGY

TOOL SUMMARY: Software requirements engineering methodology (SREM) was developed in response to continuing, and increasing, difficulties in developing complex, large, real-time software for ballistic missile defense (BMD)systems in the early 1970s. SREM is a formal step-by-step process for defining data processing requirements. It provides the means to thoroughly evaluate the adequacy of system requirements towards the goal of attaining good software specifications for any system prior to design and coding. Its goal is to reduce software development cost and schedule risk. In addition to the step-by-step requirements engineering techniques, SREM includes a machine-processsable "English-Like" requirements statement language (RSL) and a requirements engineering validation system (REVS) to automatically process the requirements statements, and to perform a wide range of analyses and simulations on its centralized data base. CONTACT: TRW, Inc., Huntsville Facility, 7702 Governors Drive West, Huntsville, Alabama 35805 USA

***FOCUS** - PROGRAM CONSTRUCTION AND GENERATION*

RESTRICTIONS: (copyrights, licenses, etc.): Licensed in-house use, information builders; usage basis, tymshare

TOOL SUMMARY: Focus is an interactive informational control system that contains facilities for describing files, for entering, changing and deleting records in files, and for reporting, graphing, modelling and statistically analyzing data from file information. Focus contains many DBMS type facilities and can access data from IBM's IMS and CuI1inane's IDMS databases as well as from focus created files. Features include: Hierarchal and relational file structures, interactive English language report writer, graphing, statistics, fle maintenance, 3270 full screen formatted data entry, financial modelling, interfaces to IMS, IDMS, V.SAM and ISAM files. CONTACT: Information Builders, Inc., 254' West 31st Street, New York, NY 10001.

***PAC III** - PROGRAM MANAGEMENT AND CONTROL*

TOOL SUMMARY: PAC III is designed to aid in the management of projects of all kinds by providing for the budgeting, planning, monitoring, and costing of all aspects of project management. It consists of nine computer programs that operate on a sequentially organized data base. Resource scheduling can be on priorities, availability, and/or network dependencies. Single or multiple projects can be scheduled as well as individual resources, group resources, or unlimited resources. Projects can be of all types and include maintenance or new developments. PAC III functions include maintenance or new developments. PAC III functions on parameters specified by the user. The user can select features, output and run frequency at run time. Includes: User's manual, implementation Guide. Contact: International Systems, Inc., 150 Allendale Road, King of Prussia, PA 19406

***TAPS/AM** - Terminal Application Processing System/Applications Manager*

RESTRICTIONS: (Copyrights, Licenses, ETC.): For lease, for sale

***TOOL SUMMARY:** TAPS/AM (Terminal Application Processing System/Applications Manager) is a support product designed to increase productivity in developing and maintaining on-line systems in various processors. The system provides routines to perform standard functions commonly programmed into most on-line applications. It drives these routines and the user application code through a generalized table structure created from information derived from input data sheets. On-line testing simulator for batch mode, screen and file recovery, and an ability to program in higher-level languages. Standardized facilities include:*

- *Sign-on/Sign-off*
- *Application Selection*
- *Transaction Menu Selection*
- *Terminal Operator Interrupt*

Automatic capabilities include:

- *Data Inquiry*
- *Collection*
- *Paging*
- *Screen Processing*
- *Data Format Editing*
- *Combination Communications Monitor*
- *Applications Manager for IBMS Systems.*

CONTACT: Decision Strategy Corp., New York, NY USA

LIBRARIAN, SOURCE PROGRAM MANAGEMENT LIBRARY

COMPUTER IBM 360/370

TOOL SUMMARY: The Librarian is a source program management system. Source programs can be stored and subsequently retrieved and updated using system commands. System facilities are included to protect against unauthorized access to master files. Programming facilities include commands for inserting, deleting and replacing source statements; syntax checking of cobol programs; editing and scanning; provisions for copying; renaming and applying temporary changes to source programs; user exits for specialized own-code interfaces; and the ability to rearrange and expand statements within a source program. Management facilities include the ability to produce reports showing the status and attributes of all source programs within a master file, including a historically accurate, date-stamped audit trail of all changes made to a program. A TSO interface option permits TSO users direct access to program modules.

CONTACT: Applied Data Research, Inc., Princeton, NJ 08540

ARTS - AUTOMATED REQUIREMENTS TRACEABILITY SYSTEM

TOOL SUMMARY: ARTS is a bookkeeping program which operates on a data base consisting of system requirements and requirement-related attributes. The major function of ARTS is to provide rapid and accurate traceability, upward and downward, in a requirements structure (TREE). Traceability allows assessment of the impact of changes, assures that top-level requirements are satisfied by the lower-level structure, facilitates generation of test plans and testing against requirements, and is essential for structured design and development. By including requirement-related attributes in the data base, automation can be extended beyond traceability. For example, schedule dates for various project events can be included, and events scheduled to occur during a specified interval can be accessed, sorted, and printed. CompIete flexibility is provided to the user in determining the attributes to be included in the data base.

Contact: Lockheed Missiles and Space Co., Inc., 1111 Lockheed Way, SunnyvaIe, CA -94086

FAME, FRONT-END ANALYSIS AND MODELING ENVIRONMENT

TOOL SUMMARY: FAME, the higher order software, inc. Fron-end analysis and modeling environment, is an interactive computer aided design tool that allows users ***to*** *build, analyze, validate, store and graphically display models of systems. Use of FAME promotes higher types of models necessary for system life cycle development and management, and insures consistency between them. The techniques employed by HOS, Inc. have been developed over a number of years with a view toward providing a complete methodology for specification of complex, large scale systems. It has effectively been used for a variety of applications ranging in size from small and simple to large real-time systems.*

CONTACT: Higher Order Software, Inc., 131 Jericho Turnpike, Jericho, NY 11753

EAVS, EXTENSIBLE AUTOMATED VERIFICATION SYSTEM

TOOL SUMMARY: EAVS is a system of compatible tools for analyzing source programs written in either the J38-2 dialect of the JOVIAL language or IBM FORTRAN IV. EAVS is intended to be applied during program testing to aid in identifying untested paths and specifying test cases that will improve testing coverage. A1l of this is provided by analysis of program structure, instrumentation of the system with software probes that measure testing coverage, and generation of comprehensive reports which pinpoint paths in the program structure that remain to be exercised. In addition, guidance is provided for the generation of test cases that will assure coverage of the untested portions. Contact: General Research Corporation, Santa Barbara, CA

GEN - A - SCREEN

TOOL SUMMARY: GEN-A-SCREEN is a software development productivity and software integration aid. By providing automation of the screen form generation process it reduces the complexity and allows a common transaction processing approach. This program standardizes systems documentation, and minimizes the impact of using different terminal configurations. Contact: Caci-Federal, Advanced Data Base Systems, 5010 Trindle Road, Mechanicsburg, PA 17055

DATAMANAGER

RESTRICTIONS: Licence

TOOL SUMMARY: Datamanager is a data dictionary, and generally regarded as an aid to the data administration functions. It has also been applied to business systems planning and documentation areas. A full implementation of data resource management provides ease of use, flexibility, and the enforcement of standards. Datamanager is referred to as a nucleus unit with range of selectable units to select the exact *configuration desired. The nucleus provides definition support, query, top-down structure and use, creation/generation, and error recovery. The dictionary affords full procedural and historical information, answers to "what if" question preceding a change, removal of specified data items and management of the inventory if an organization process and data resources scheme. Datamanager interfaces with these* ***DBMS = ADABAS, IDMS, IMS (DL/1),*** *Total, System 2000, programming.*

APPENDIX A

A GUIDE TO THE CONTENT OF SPECIFIC RECOMMENDATIONS

Chapter 1

1-A *Explore productivity enhancements to aid in producing required software for integrating CAD/CAM process: theme - tools and the knowledge to use them.*

1-B *Seek standardization of software tools to improve Navy-Industry and intra industry automated interface of integrated CAD/CAM system.*

1-c *Maintain effective contact with governmental agencies and other industries to enable a compilation. of new CAD/CAM technologies potentiality applicable to shipbuilding in order that a baseline of anticipated software needs can be compiled.*

Chapter 2

2-A *Recognize software has no clear focus, but is nonetheless a pivotal productivity and cost issue for the design/production integration process.*

2-B *Advocates support of the CAD/CAM integration process through use of software tools to improve productivity of the required software.*

2-C *Points to the subtle role of software productivity in shipbuilding CAD/CAM integration and emphasizes the overt result of value; which . is the bringing on-line of modern CAD/CAM equipments much quicker and at lower cost.*

Chapter 3

3-A *Emphasizes the synergistic effects of CAD/CAM integration to completely redefine the scope of required software needs in the shipyard environment.*

3-B *Points to the similarity of shipbuilding to the discrete batch manufacturing classification of industry; an area of great difficulty for CAD/CAM applications, but also one of many rewards.*

3-c *Advocates selection of qualitative features of CAD systems to increase ability to do design tasks and cautions against the proliferation of CAD workstations having limited capabilities.*

3-D *Indicates that NC machining is economical for jobs requiring small output quantities, and especially useful for making parts which frequently undergo. design changes.*

3-E *Suggests that existing organizational settings for CAD and CAM systems must undergo in industry changes to enable effective CAD/CAM integration.*

3-F *Points to the need to have a new class of system engineer redefine CAD/CAM integration for the shipbuilding setting to properly examine all variables arising from the synergistic effects of integration.*

Chapter 4

4-A *Points to the pressing need for systematic analysis of CAD/CAM technologies in other industries as a basis for defining feasible and flexible systems: technological transfer via scenarios is the suggested method.*

4-B *Stresses the value. of developing future shipyard environments as a prequisite to analyzing alternate approaches to CAD/CAM integration, and related "productivity improvement methods.*

4-c *Advocates use of several future shipyard environments to minimize deleterious effects of misjudging future technological trends.*

4-D *Suggests a formal technology transfer approach to accomplish development of a means to handle the large quantities of data that must be reviewed to create viable integrated CAD/CAM systems from which to develop requirements for computer software.*

4-E *Indicates the need to consider creation of software in the tangential areas of MIS and related inventory control areas to accommodate opportunities arising out of integrated CAD/CAM functions.*

4-F *Highlights the wide diversity of CAD/CAM integration approaches available in other industries, yet caution is urged to not transpose technologies without using some frame-of-reference to shipbuilding."*

Chapter 5

5-A *Advocates establishment of sound system definition and system integration practices to achieve reliable software requirements for integrated CAD/CAM systems: software engineering and an understanding of the software life cycle is central to this goal.*

5-B *Suggests that attributes of Computer-Based Systems are similar to a degree sufficient to enable software tools for CAD and CAM use to be productively used for many other in-house software needs.*

5-c *Indicates that underlying software issues are often the most costly component of software projects.*

5-D *Urges systematic efforts to plan software tool introduction in concert with upgrades in hardware.*

.5-E *Points out that protracted maintenance/upgrade of existing software on older computers becomes non-productive well before hardware capacity is reached.*

5-F *Advocates the development of a shipyard indigenous software life cycle recognizing that there are many varied approaches to select from.*

5-G *Encourages the linking of both technical and management needs through the use of software tools.*

5-H *Outlines the basic differences between hardware and software projects, and why software tools are important to the entire life cycle.*

5-I *Encourages examining the basic similarities of computer-based systems as a means of encompassing overall software development in CAD/CAM and MIS areas to afford the option of extra capabilities at minimal cost.*

Chapter 6

6-A *Suggests that shipyards subject modernization policy to proven shipbuilding methodologies (ZOPM) and utilize changes these practices will bring to the yards as drivers for selecting CAD/CAM systems: software to integrate these systems will give a truer picture of need than software required to integrate existing identified "islands of automation".*

6-B *Advocates clear definition of shipyard modernization policy prior to selection of CAD/CAM equipments - this being a critical part of the system definition process.*

6-C *Indicates the ability of outfit planning concepts of information handling to form the basis of computer data base systems - this approach being desirable over force fitting shipyard data to the dictates of a packaged software system.*

6-D *Urges management of integration programs to establish controls for the integration process.*

Chapter 7

7-A *Emphasizes the importance of arrangements which provide for coordinating the contribution of new technology, system definition, and software productivity enhancements to support shipyard modernization activities: use of a disciplined systems integration plan is suggested.*

7-B *Suggests attention to long-range pIanning of yard update activities in borad terms to give a framework for near-term building of a systems integration plan.*

7-c *Proposes the creation of goals for personnel involved in CAD/CAM integration - with software tools an important communication medium tube used by all participants.*

Chapter 8

8-A *Proposes use of a narrative similation process, based on a shipyard scenario, to aid in formulation of the intricacies of using new CAD/CAM technologies in the shipyard this as an aid to elicit a software - tool plan (as well as other goals).*

8-B *Points to the need to categorize software tasks in a way to enable projecting the benefits of system integration to assess porential economics of joint departmental funding of software tasks - software tools are indicated to be a catalyst to make these economies possible.*

8 - c Suggests a method of presenting the findings of analysis of proposed CAD/CA,M systems in a manner to allow evaluation by a wide range of shipyard personnel and subject matter experts.

Chapter 9

9-A Encourages managers and technical personnel to cooperate in developing and using, software cost estimating methodologies indigenous to the shipyard environment to determine accurate cost and cost benefits of different software development approaches, and savings of different software tools.

9-B Suggests familiarity with a wide-range of cost estimating methods to enable selection of useable software cost models, and proper insight into their results and limitations.

9-C Emphasizes the use of software cost estimating techniques as a means to evaluate cost savings of software tools, and thus permit calculation of their productivity improvement potential

Chapter 10

10-A Outlines classification schemas of software tools, their availability/sources and guides users to indexing and abstracting services, as well as specific vendors, of software tools.

10-B Suggests a means of matching software tools to a shipyard software development process through classification of software phases into tasks and functions - which permit a match with a classification scheme of software tool functions.

lo-c Urges the development of many different sources of software tools to enable both the regular review of new technologies and permit timely *use of tools as required by developments in the shipyard CAD/CAM integration environment.*

APPENDIX B

CHRONOLOGY AND OBJECTIVES OF INTERIM PRESENTATIONS

Gathering of data for the Software Tools for Shipbuilding Productivity report depended largely on visits to shipyards. Nine different shipyards were visited. some more than once in the course of this study. The data gatherred was presented to assembled groups of shipbuilders and ship designers periodically during the report preparation period to enable a review of progress, and seek opinions on further activities.'

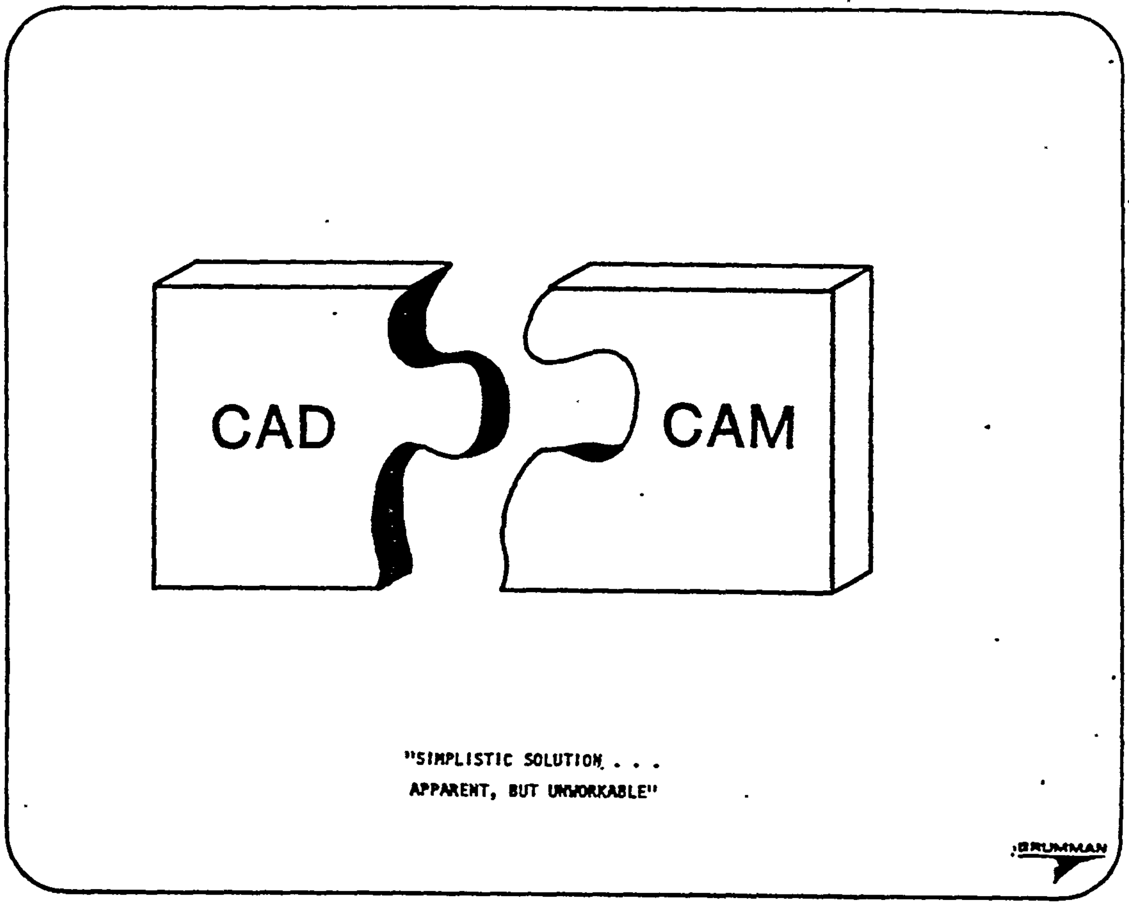
June 21, 1983 *SP-4 Meeting* *Sturgeon Bay, Wis.*	*Progress Report No. 1; definition and use of software tools; theme of report: "Tools and Knowledge to Use Them" adopted.*
June, 1983 *CAD/CAM Advisory Panel* *Chicago, III.*	*Review of interface between IIT Research Institute's CAD/CAM survey and software tools project. Joint on-site visits to selected sites were planned.*
August 30,,1983 *SNAME AD HOC PANEL on Computer-Aided Ship Design*	*Presentation of project progress and goals; interface with ship design system goals.*
October 5, 1983 *SP-4 Meeting* *Long Beach, Ca.*	*Progress Report No. 2; Overview of CAD/CAM integration scenarios and impact of CAD on integration needs.*
February 14, 1984 *SP-4 Meeting* *Moorestown, NJ*	*Progress Report No. 3; Technology transfer findings elicited from on-site visits and scenario data analysis.*
May 10, 1984 *SP-4 Meeting* *Brunswick, Maine*	*Report completion data and planning of industry presentation format, time, and place.*
October 23, 24, 1984 *CAD/CAM Seminar and SP-4 Meeting* *Ann Arbor, Mich.*	*A day-long seminar entitled: "Software Tools for Integration of the Shipbuilding Design/Production Process" held at the University of Michigan. An overview of specific report recommendations and papers by experts in CAD/CAM covering these issues made up this industry presentation.*

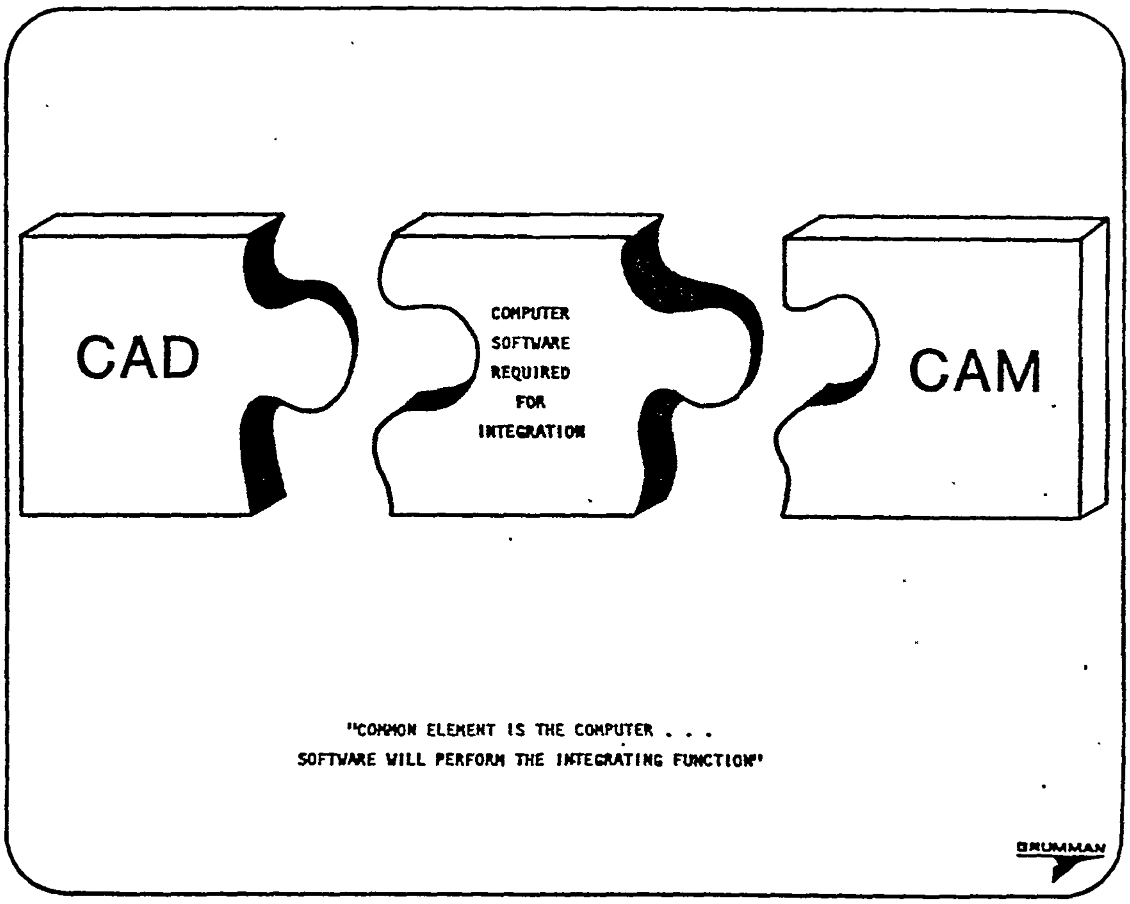
APPENDIX C

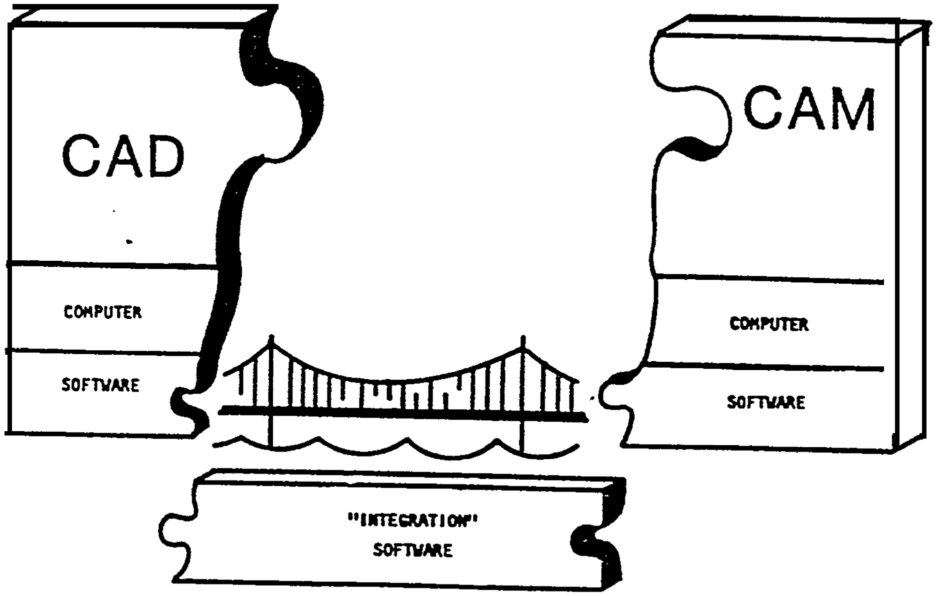
SOFTWARE TOOLS: A BRIEF PRESENTATION

Software Tools for Shipbuilding Productivity

SOFTWARE TOOLS FOR INTEGRATION OF THE SHIPBUILDING DESIGN/PRODUCTION PROCESS

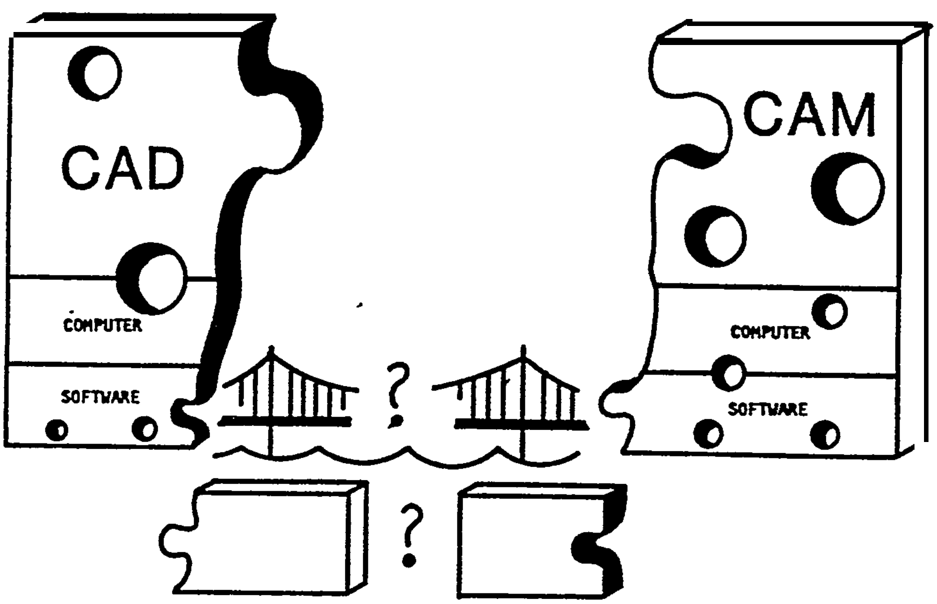

CAD
CAM
"SIMPLISTIC SOLUTION . . .
APPARENT, BUT UNWORKABLE"
GRUMMAN


CAD
COMPUTER
SOFTWARE
REQUIRED
FOR
INTEGRATION
CAM
"COMMON ELEMENT IS THE COMPUTER . . .
SOFTWARE WILL PERFORM THE INTEGRATING FUNCTION"
GRUMMAN



THE SOFTWARE WILL "BRIDGE THE GAP"
BETWEEN CAD/CAM

GRUMMAN



SINCE THE "REAL WORLD" CAD/CAM SYSTEMS
ARE INCOMPLETE, DESIGN/PRODUCTION
INTEGRATION CANNOT PROCEED UNTIL
THESE "HOLES" ARE FILLED.

GRUMMAN

ISSUES

CAD/CAM SOFTWARE IS THUS IMPOSSIBLE TO COMPLETE, AS THE "THINGS" IT INTEGRATES ARE:

- o NON EXISTING
- o CONTINUOUSLY CHANGING
- o "DIFFERENT" FROM YARD-TO-YARD THROUGHOUT THE SHIPBUILDING COMMUNITY

THE CONTINUOUS PROCESS OF UPDATING INTEGRATION SOFTWARE ON AN AD HOC BASIS WOULD BE:

- o EXPENSIVE
- o TIME-CONSUMING
- o SUSCEPTIBLE TO TECHNOLOGICAL OBSOLESCENSE

GRUMMAN

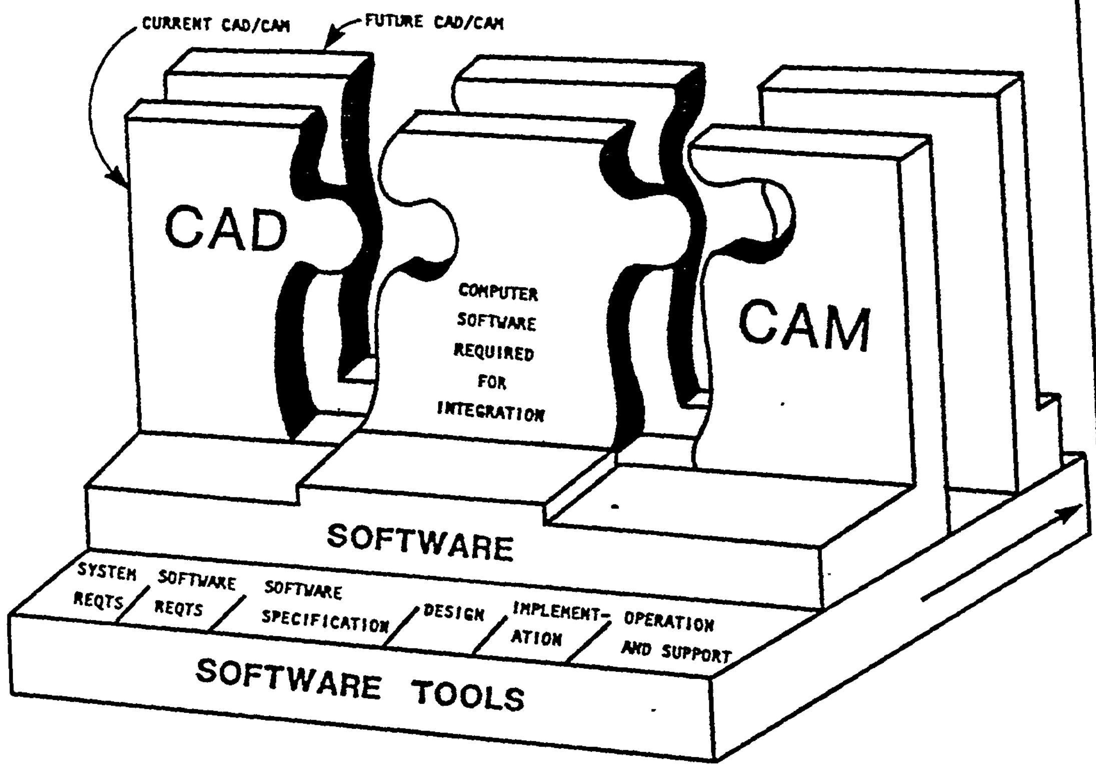
SOLUTION

- o RECOGNIZE THIS "STATE OF SOFTWARE FLUX" AS THE IMPEDIMENT TO CAD/CAM INTEGRATION

- o UTILIZE MODERN SOFTWARE TOOLS AS AN APPROACH TO DEAL WITH:
  - ALL SOFTWARE DEVELOPMENT STAGES
  - ALL CAD SOFTWARE REQUIRED
  - ALL CAM SOFTWARE REQUIRED
  - ALL "INTEGRATION" SOFTWARE REQUIRED

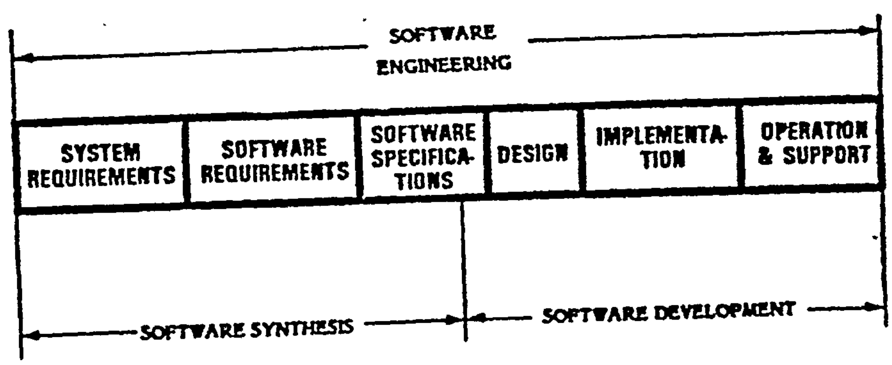
- o CAPITALIZE ON USE OF SOFTWARE TOOLS TO EFFECTIVELY MANAGE THE SOFTWARE LIFE-CYCLE

GRUMMAN

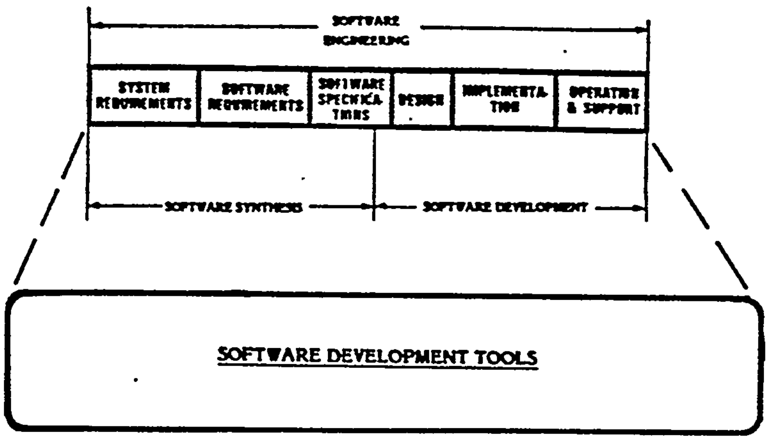


SOFTWARE TOOLS & SYSTEM GROWTH

CAD/CAM SOFTWARE ENVIRONMENT

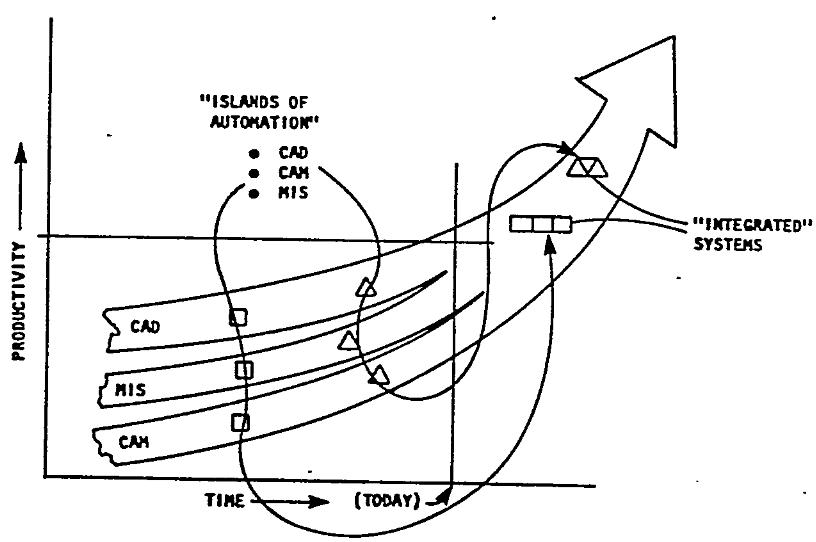


CAD/CAM SOFTWARE ENVIRONMENT



- SYSTEM DESIGN METHODOLOGIES
- ANALYSIS TOOLS
- TEST SUPPORT TOOLS
- CONFIGURATION MANAGEMENT TOOLS
- FACILITIES SUPPORT TOOLS
- MANAGEMENT TOOLS
- DOCUMENTATION AIDS
- COMMUNICATION SYSTEM TOOLS

GRUMMAN



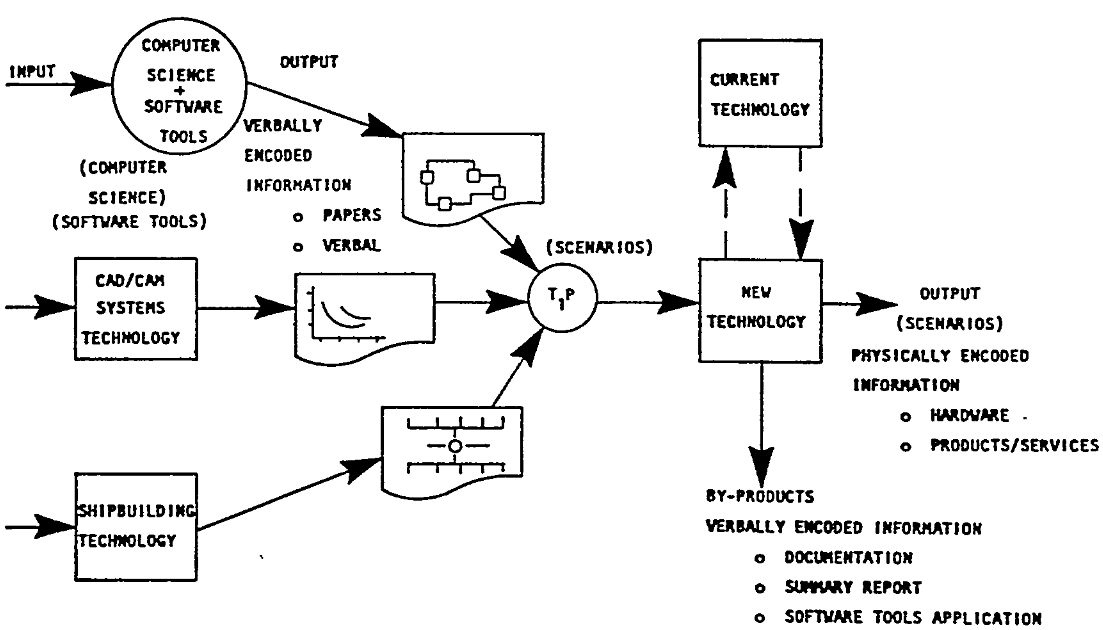
INTEGRATION OF CAD/CAM/MIS "ISLANDS OF AUTOMATION" TO AFFECT FUTURE PRODUCTIVITY ENHANCEMENTS

GRUMMAN

SCENARIOS

ALTERNATIVE DESCRIPTIONS OF WHAT THE FUTURE MIGHT LOOK LIKE, BASED UPON INFORMATION GATHERED AND ANALYZED BY OTHER MEANS.

GRUMMAN



USE OF SCENARIOS TO DEVELOP RELEVANT SHIPBUILDING CAD/CAM INTEGRATION FORECASTS

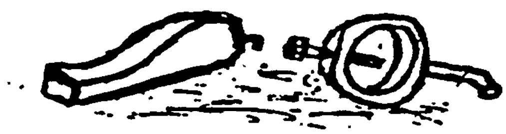
GRUMMAN

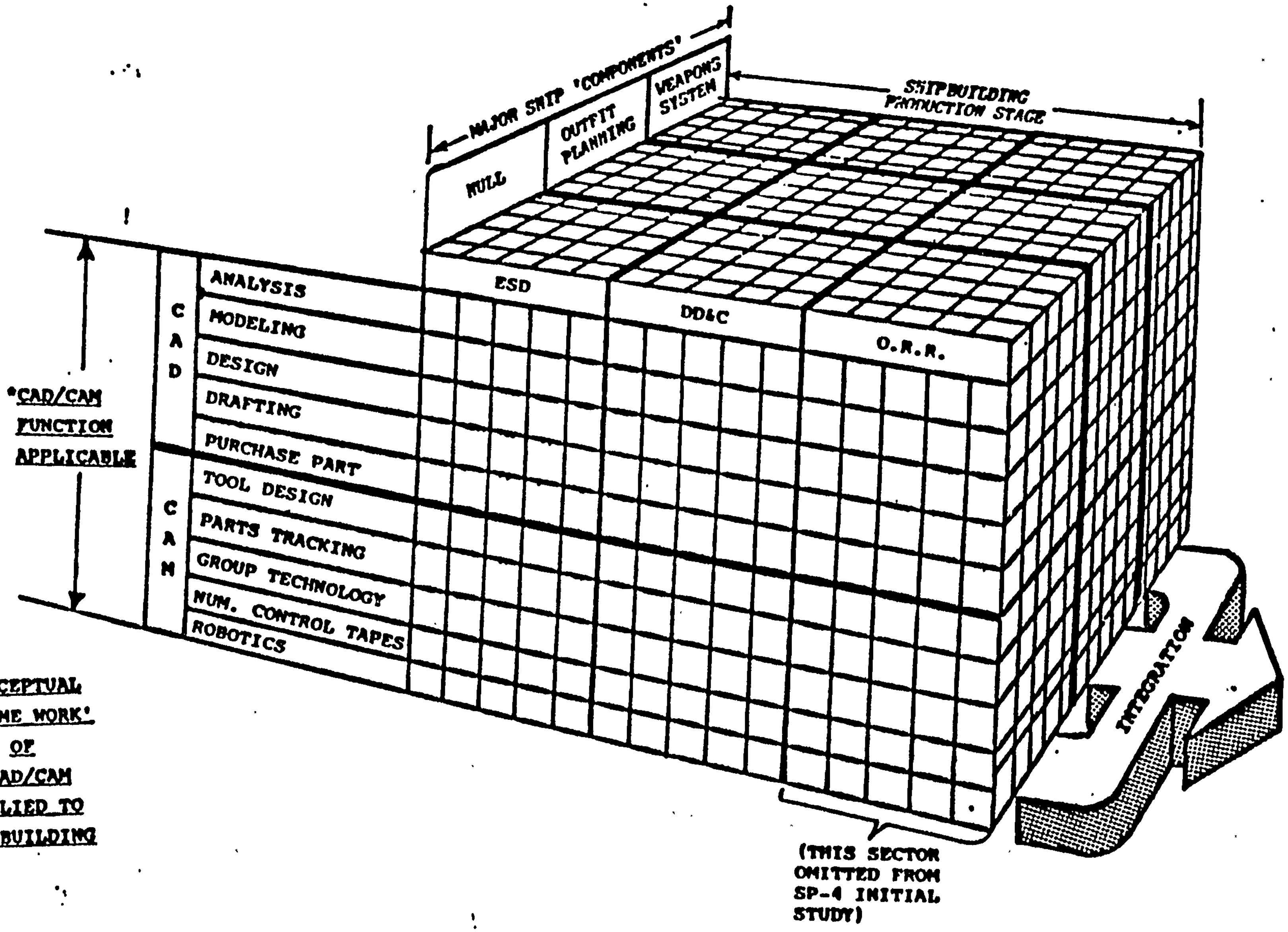
CAD/CAM: A SYSTEMS APPROACH

CAD/CAM SCENARIOS	I. **COAL DEFINITION/PROBLEM STATEMENT**
	2. **OBJECTIVES AND CRITERIA DEVELOPMENT**
SOFTWARE ENVIRONMENT	3. SYSTEMS **SYNTHESIS**
	4. **STSTEMS ANALYSIS**
	5. **SYSTEMS SELECTION**
	6. **SYSTEM IMPLEMENTATION**
	7. **SYSTEMS OPERATION AND SUPPORT**

GRUMMAN

Tools and the Knowledge **to Use Them**





CONCEPTUAL 'FRAME WORK' OF CAD/CAM APPLIED TO SHIPBUILDING

DEFINED SCOPE OF STUDY

GRUMMAN

APPENDIX D

ACRONYMS USED

APT	*Automated Programming of Tools*
AVS	*Automated Verification System*
CAD	*Computer-Aided Design*
CAE	*Computer-Aided Engineering*
CAM	*Computer-Aided Manufacturing*
CNC	*Computer Numerical Control*
COMIC	*Computer Management and Information Center*
CRT	*Cathode Ray Tube*
DBMS	*Data Base Management System*
DD&C	*Detail Design & Construction*
DDS	*Data Directory System*
DED	*Data Element Dictionary*
DNC	*Direct Numerical Control (also Distributed Numerical Control)*
ESD	*Early Ship Design*
FM	*Frequency Modulation*
FMS	*-Flexible Manufacturing System*
GAO	*Government Accounting Office*
HBCM	*Hull Block Construction Method*
HID	*Hierarchical Item Descriptor*
HIPO	*Hierarchical Input/Output (Charts)*
HW	*Hardware*
ICD	*Interface Control Document*
ICST	*Institute for Computer Sciences and Technology*
1/0	*Input - Output*
IPS	*Iterations Per Second*
IT&E	*Integrated Test and Evaluation*
LCC	*Life Cycle Cost (Concept)*
LOC	*Lines of Code*

MAC	Management/Maintenance and Control (Tools)
MarAd	Maritime Administration
MODEM	Modulator/Demodulation
MRP	Material Requirements Planning
NBS	NationaI Bureau of Standards
NC	Numerical Control .
NTIS	National Technical Information Service
OR&R	Overhaul, Repair, and Refurbishment
OT&E	Operational Test and Evaluation
P c	Process Controller
PPFM	Pipe Piece Family Manufacturing
PSL/PSA	Problem Statement Language/Problem Statement Analysis
PWBS	Product Work Breakdown Structure
QA	Quality Assurance
QC	Quality Control
RAD	Requirements Analysis and Design (Tools)
REVS	Requirements Engineering Validation System
SAM	Software-Simulation and Modeling (Tools)
SCES	Software Cost Estimating System
SIP	System Integration Plan
SM	Solid Modeling
SP-4	SNAME Ship Production Committee No. 4 (SP-4) on Design/Production Integration
SREM	Software Requirements Engineering Methodology
TF	Technological Forecasting
TTP	Technology Transfer Process
VHLL	Very High Level Language
WBS	Work Breakdown Structure
ZOPM	Zone Outfit Planning Method
ZPTM	Zone Painting Method

APPENDIX E

CHAPTER REFERENCES

Chapter 1

1. *Diesslin, R.L., A Survey of CAD/CAM Technology Applications in the U.S. Shipbuilding Industry (1983), IITRI project K06011; IIT Research Institute, Chicago, IL. (January 1984), pp. 2, 6, 122.*

2. *Fowler, E.B., "NAVSEA CAD/CAM Project Organization Alternatives". Enclosure -33 pps. - cover letter dated 20 April 1981 to Chairman Ship Productivity Committee (June 3-5 1981, SP-4 meeting, Atlanta, GA.), pp. 6, 13.*

3. *SP-4 "CAD/CAM Integration Sub-Panel, 3. Lina Chairman". Notes from SP-4 Sub-Panel Meetings, Atlanta, G.A., (June 4-5 1981), pp. 1-7.*

Chapter 2

1. *Committee on Navy Shipbuilding Technology, Marine Board and Commission on Engineering and Technical Systems, National Research Council. Productivity Improvements in U.S. Naval Shipbuilding, National Academy Press, Washington, D.C. (1982), pp. 3,44, 50.*

2. *Doetsch, E. C., "CAM: 'Missing Link Between Management and Production Line". Electronic Design, (7 January 1982), pp. 184.*

3. *Fowler, E.B., op. cit. pp. 13-14.*

4. *SP-4 "Meeting Notes". From SP-4 Design/Production Integration Panel of National Shipbuilding Research Council Meeting, Atlanta, GA. (23-24 May 1982), pp. 3-5.*

5. *Wiedenhaefer, P., "The Engineering Production Integration Process: Graphics, Interactive Computing and Data Base". In Proceedings of the Shipbuilding Design/Production Integration Workshop - Vol. I, Atlanta, GA. (January 18-21 1981), pp. 21.*

Chapter 3

1. *Allan, R., "Technology Forecast: Industrial Electronics (1982)". Electronic Design, (7 January 1982), pp. 94-118.*

2. *Blackburn, J.E., In European Scientific Notes (ESN 37-7; 31 July 1983) Office of Naval Research, London from Proceedings of the Third International Conference on Engineering Software, Imperial College, London, England (April 1983), pp. 254-256.*

3. Borrell, J., "Solids Modeling Becomes the Driving Force Behind Automation" Digital Design (November 1983), pp. 88-98.

4. "CAD/CAM - The Factory Integrator" (No author) - Production Engineering (April 1983), pp. 65-67.

5. Folkman, J., "DNC. What Is It?" Tooling and Production (January 1982), pp. 70-73.

6. Inglesky, T., "Getting to the Heart of CAD/CAM". Assembly Engineering (June 1982), pp. 46-47.

7. Inglesby, T., "New CAD/CAM Entries Have Designs on the Future". Infosystems (November 1982), pp. 56-62.

8. Krouse, J. K., "CAD/CAM - Bringing the Gap from Design to Production". [Machine Design (12 June 1980), pp. 117-121.

9. Krouse, J. K., "CAD/CAM Broadening its Appeal". Machine Design (22 April 1982), pp. 54-59.

10. Marks, L., "CAD Systems Update". Electronic Packaging and Production (May 1982), pp. 55-70.

11. Tuck, J., "Successor to CAD/CAM". Circuits Manufacturing (February 1984), pp. 16.

12. "Use of Numerically Controlled Machine Tools Aboard U.S. Navy Destroyer Tenders". Defense Technical Information Center. DLA. Cameron Station, Alexandria, VA. (February 1970), pp. 3-10, 21-28.

13. Morrison, L., "CAD: Snake Oil or Cure All?" Plan and Print (May 1983), pp. 22-23.

14. Baker, D.S., "CAD Angles...", Plan and Print (February 1974), pp. 6-7.

15. Singer, P., "Mid-Range CAD/CAM/CAE Systems Dominate Graphics Display Terminals". Mini-Micro Systems (July 1984), pp. 223-228.

16. Sternberg, J., "CAD/CAM - More Than a Drawing Board...Less Than a Miracle...a Revolution in Industry". Product Design and Development (July 1983), pp. 22-29.

Chapter 4

1. Albert, M., "Fixturing Keeps FMS on Track". Modern Machine Shop (March 1984), pp. 50-59.

2. Allan, R...Op. Cit., pp. 94-118.

3. Anton, P., P.L. Arthur, and J.W. Duncan, Technological Forecasting: Tools, Techniques, Applications. Report No. 115. AMA Bulletin, New York, New York Research and Development Division (1967), pp. 7-9, 24, 26.

4. Barkmeyer; E., C. McLean, and M. Mitchell, "A Computer Architecture for Small-Batch Manufacturing". National Bureau of Standards in IEE Spectrum (May 1983), pp. 59-64.

5. Bright, J.R., Technological Forecasting for Industry and Government: Methods and Applications. Englewood Cliffs, N.O., Prentice-Hall, Inc. (1968), pp. 146-149, 153-157, 210-211, 270-275, 359, 439-433.

6. General Electric Company; "Factory with a Future Products and Services". Automation Planning Guide - a Step-by-Step Approach to Automation (1980 - G.E. Technical Brief).

7. Grumman Data Systems "Distributed Numerical Control". Corporate Presentation Material (April 1982).

8. Hart, J.L., "Technological Forecasting from Board Room to Drawing Room". Machine Design (12 February 1976), pp. 90-93.

9. Martino, J.P., Technological Forecasting for Decision .Making. New York, New York, American Elsevier, Inc. (1972), pp. 65-69, 287-295,331-343.

10. Olenzak, A.T., "Technological Forecasting: A Programmatic Approach". Chemical Engineering Progress, Volume 68, No. 6. (June 1972), pp. 27-32.

11. Pouder, R. and R. Skirkanich, Computer Automated Shipbuilding Technology. Grumman Data Systems Corporation Overview Report (24 April 1981), pp. 21-23, 38-40.

12. Ware, R. B., "Systematics: What a Systematic Needs to Known and Do". Infosystems (May 1978), pp. 106.

13. Zdeblick, W.J., "Process Design by Computer". IEEE Spectrum (May 1983), pp. 56-58.

Chapter 5

1. Allan, R...op. cit., pp. 96, 109.

2. Alford, M.W., and J.T. Lawson, Software Requirements Engineering Methodology (Development). Report No. RADC-TR-~9-16s, Rome AFB, New York (1979), pp. 4-6, 24-31, 70-75.

3. Belady, L.A. and M.M. Lehman, "A Model of Large Program Development", IBM Systems Journal (No. 3-1976), pp. 225-227, 234-240.

4. Chen, R. and E. Cuthill, Computer Science Research and Development in Support of Ship Design Production and Repair (1978), Office of Naval Research Report by David W. Taylor Naval Ship Research and Development Center, Bethesda, Md., pp. 5, 8-11, 62-63.

5. Committee on Navy Shipbuilding Technology, Op. Cit., pp. 6, 43, 51.

6. Department of Defense (DoD) Directive, Management of Computer Resources in Major Defense Systems. No. 5000.29 (April 26, 1976), pp. 1-2.

7. Hagan, S. R., An Air Force Guide for Monitoring and Reporting Software Development Status. No. AD-AO16-488, Air Force Systems Command, Hanscom AFB, Bedford, Mass. (1975), pp. 8-12,57-66,78-82.

8. Hymowitz, C., "Manufacturers Press Automating to Survive, But Results are Mixed". Wall Street Journal (March 11, 1983), pp. 1.

9. Keller, E. L., "Industrial Automation: Awaiting the Final Link - Standards". Systems and Software (June 1984), pp. 96-97.

10. Kull, D., "To Raise Productivity Work Smarter, Not Harder". Computer Decisions (March 1984), pp. 166-169.

11. Langler, G.H., "It's Time for Computer Integrated Engineering", Test and Measurement World (January 1984), pp. 54-56.

12. Maritz, P., "Software Development". ,Mini-Micro Systems (December 1983), pp. 211, 213, 216.

13. Morgan, B. and R. Skirkanich, "Large Scale Software Design Management Systems: Application Study and Implementation" from AIAA Computers in Aerospace Conference, Los Angeles, Calif. (1979), pp. 1-9.

14. Naval Electronic Systems Command, Computer Software Life Cycle Management Guide. NAVELEXINST 520023 (Computer Resources Management Guidebook Series for NAVELEX Project Managers - 1 March 1979), Sections/pp. I; 1.2, 3.1: II; 2.1, 3.o, 4.1.1,5.1.2.

15. Nelson, R.L., "Increasing the Factors for CAD/CAM Growth". CAD/CAM Technology (Winter 1982), pp. 25-27.

16. Peper, P., "Implementing CAD/CAM". Manufacturing Operations (Winter 1984), pp. 26-27.

17. Perkins, D. R., P.G. Sorenson, J.P. Tremblay, and E.D. Wig, "A Survey of Some Automated Aids for Systems Analysis and Documentation". IN FOR Vol. 19, No. 3 (August 1981), pp. 205-207.

18. Rachowitz, B. and G. Sandier, "Software Cost Estimating...A Management View". Grumman/University Technology Forum; 22-23 October 1979.

19. Ruegesgger, T.B. "Software Aspects of Factory Machine Control", Computer Design (April 21, 1983), pp. 131-137.

20. Thayer, R.H., Notes from Panel Talk-on Software in U.S. Air Force, Sacramento Air Logistics Center, Calif. (AIAA Computers in Aerospace Conference, Los Angeles, Calif. 1979).

Chapter 6

1. Activity Roadmap for SP-4 Software Tools Project: Progress Report No. 1; 21 June 1983; Sturgeon Bay, Wis.

2. Analysis of Critical Issues in the U.S. Shipbuilding Industry, A report to the National Center for productivity and quality of working life, by Pugh-Roberts Associates, Inc., (16 January 1978), pp. 1-28.

3. Chirillo, L.D., SP-2 Outfitting and Production Aids, a Presentation to the IREAPS Technical Symposium, Boston, Mass. (25 August 1983), pp. 1-8.

4. Chirillo, L. D., SP-2 Production Techniques, a Presentation to the IREAPS Technical Symposium, Baltimore, Md., (15 September 1981), pp. 1-5.

5. Committee on Navy Shipbuilding Technology...op. cit., pp. 3-7, 48-50.

6. Diesslin, R. L., A Conceptual Information Model...for Outfit Planning , IITRI Project K06008; IIT Research Institute, Chicago, II., September 23, 1982), pp. 3-19, 31-51, 57-66.

7. Diesslin, R.L., A Survey of...op. cit., pp. 12-19,83-89, 109-110.

8. Fowler, E.B., op. cit., pp. 6-10.

9. Graves, Dr. R.J. and Dr. L.F. McGinnis, Jr., A Decision Support System for the Outfit Planning Problem: Modeling and Concep- Report prepared for Dept. of Commerce, No. 2 (February 1980), pp. 1-63.

10. *Riesenfeld, R.F., Recommendations for Computer Utilization in Shipbuilding. Dept. of Computer Science, University of Utah. Report No. (ONR) 0001477 C0157 (21 November 1978), pp. 8-29.*

11. *SP-4 "CAD/CAM Integration Sub-Panel...", op. cit., pp. 1-7.*

Chapter 7

1. *Analysis of Critical Issues in the Shipbuilding Industry... op. cit., pp. 4-7, 12-22.*

2. *Chirillo, L.D., op. cit (1981), pp. 1-5.*

3. *Chirillo, L. D., op. cit. (1983), pp. 1-8.*

4. *Committee on Navy Shipbuilding Technology...op. cit., pp. 4-6, 48-50.*

5. *Graves and McGinnis... op. cit., pp. 12-63.*

6. *Ramsey, R., An Overview of the National Shipbuilding Industrial Base, paper presented at the 19th Annual Technical Symposium of the Assoication of Scientists and Engineers of the Naval Sea Systems Command, Wash., D.C. (April 1982), pp. 11-26.*

7. SP-4: "CAD/CAM *Integration Sub-Panel...", op. cit., pp. 1-5.*

8. *Risenfeld, R.F...op. cit., pp. 11-29.*

Chapter 8

1. *Allan, P...op. cit., pp. 94-101, 111-118.*

2. *Barkmeyer, E...op. cit., pp. 60-63.*

3. *Chen, R. and E. Cuthill...op. cit., pp. 8-10, 56-63.*

4. *ChiriIlo, L.D...op. cit. (1981), pp. 3-5.*

5. *Chirillo, L.D...op. cit. (1983), pp. 2-8.*

6. *Committee on Navy Shipbuilding Technology,...op. cit., pp. 3-7, 47-50.*

7. *Doetsch, E.C...op. cit., pp. 184.*

8. *Folkman, J...op. cit.,pp. 70-73.*

9. *Ramsay, R., Approaches to Improving Shipbuilding Productivity, paper presented at the 20th Anniversary Technical Symposium of the Association of Scientists and Engineers of the Naval Sea Systems Command, Wash., D.C., (March 1983), pp. 8-21.*

10. *SP-4: "CAD/CAM Integration Sub-Panel...", op. cit., pp. 1-7.*

11. *Wiedenhaefer, P...op. cit., pp. 16-21.*

Chapter 9

1. *Bangs, A., G. Foreman, J. Green, E. Rodriguez, R. Waina. Predictive Software Cost Model Study, Avionics Laboratory of U.S. Air Force, Wright Patterson AFB, Ohio, June 1980, pp. 21-25, 36-45, 101-107.*

2. *Belady, L.A...op. cit., pp. 232-3,238-244,249-250.*

3. *Chen, R...op. cit., pp. 54-61.*

4. *Cotton, IA. "Methodologies for the Cost-Benefit Analysis of Computer Graphics Systems" (based on NBS Technical Note 826 by I. Cotton). Computers and Graphics, Vol. I (1975), pp. 33-43.*

5. *Deshler, D. and D.L. Winchester. "Automated Tools for the Analysis and Design of Real-Time Software." Reprint, 6 pps., Hughes Aircraft Co. (1983), pp. 1-6.*

6. *Gitto P., Economical and Viable CAD/CAM, DNC Systems for Large and Small Corporations, paper presented at 14th AMTC (March 13-16, 1977), pp. 1-12.*

7. *Kull, D...op. cit., pp. 97, 108.*

8. *Morgan.. "Large Scale Softare Design .Management Systems...", op. cit., pp. 3-7, 9.*

9. *Naval Electronic Systems Command. Computer Software Life Cycle Management...op. cit., pp. II; 2.1-3.0.*

10. *Naval Ocean Systems Center and Armed Forces Communications and Electronics Association, NOSC/AFCEA 3 Intelligence Symposium Software Tools and Productivity, (Proceedings) 16-18 November, 1982; pp. session 2.*

11. *Rachowitz B. and G. Sandler. Software Cost Estimating...an A Management View. Paper from Grumman/University Technology Forum (22-23 October 1979), pp. 1-14.*

12. *Thibodeau, R. An Evaluation of Software Cost Estimating Models, Rome Air Development Center, Griffiss AFB, N.Y. - Report No. CR-1-940 (June 1981), pp. 14-20, 412-15.*

Chapter 10

1. *AIAA Computer Systems Committee Software Tools Survey (Volumes 1 and 2, 1980), Rome Air Development Center, U.S. Air Force Systems Command, Griffiss Air Force Base, N. Y., pp. 1-315.*

2. *Chen and Cuthill...op. cit., pp. 9-11, 54-53.*

3. *Hagan, S. R...op. cit., pp. 8-12, 78-80.*

4. *Houghton, R. C., Jr., Software Development Tools, (1982). National Bureau of Standards Special Publication 500-88, pp. 1-193.*

5. *Kull, D...op. cit., pp. 166-169.*

6. *Naval Electronic Systems Command. Computer Software Life Cycle Management Guide... op. cit., pp. I; 1.2-3.1: II; 2.1-4.1.1.*

7. *Perkins, D.R...op. cit., pp. 205-207,217-218.*

8. *Schindler, M., "Software Automation Attacks the Programming Bottleneck". Systems and Software...Editorial Series Reprint for the Systems Integration Reprinted from 12 November 1981, Electronics Design, pp. 11-16, 33-35.*

APPENDIX F

FIGURE REFERENCES

CHAPTERS AND FIGURE SOURCES: (NO FIGURES)

CHAPTER 3 FIGURE SOURCES:

Figure 3.4.1 *CAD System Elements (Prepared for Report).*

Figure 3.6.1 *CAD/CAM Interface Matrix (Prepared for Report).*

Figure 3.6.2-1 *Design/Production Data Flow (Adapted from AlIan, R., Inglesky, T., and "CAD/CAM - The Factory Integrator").*

Figure 3.6.2-1 *CAD/CAM Synergy Computer Architecture - (Adapted from Krouse, J., Inglesky, T., and Allan R.).*

CHAPTER 4 FIGURE SOURCES:

Figure 4.1 *Forecasting Techniques: Conceptual Schematic (Olenzak, A.T.).*

Figure 4.1-1 *Technology Transfer Process (Adapted from Anton et. al., and Bright).*

Figure 4.1-2 *Components of Technology Transfer Process (adapted from Anton et al., and Bright).*

Figure 4.1-3 *Use of Scenarios for Technology Transfer in Shipbuilding (compiled for report).*

Figure 4.2 *Production Operations Overview (adapted from IBM Manufacturing Industry Marketing and Ramsey, 1982).*

Figure 4.2-1 *Integrated CAD/CAM Component Subsystems (General Electric co.).*

Figure 4.2-2 *Flexible Manufacturing System (adapted from,Albert, M.).*

Figure 4.2-3 *Concept of Hierarchy for CAD/CAM Computers (adapted from Allan, R.).*

Figure 4.2-4 *Integrated CAD/CAM Command Nomenclature (Barkmeyer, E.,et al.).*

Figure 4.2-5 *Distributed Numerical Control - DNC (Grumman Data Systems, April 1982).*

Figure 4.2-6 *Integrated CAD/CAM Process Planning (Zdeblick, W.J.).*

Figure 4.2-7 *Activity Structure for New Generation Shipbuilding Practice (Pouder, R. et al. as adapted from Chirillo, L.D. -1981. and 1983).*

CHAPTER 5 FIGURE SOURCES:

Figure 5.1-2 *Evolution of Software Requirements (adapted from Hagan, S.R. and interview notes).*

Figure 5.2-2 *History of Software Engineering (adapted from 1979 AIAA Computers in Aerospace Conference Notes).*

Figure 5.3-1 *Module with Examples of Assigned Resources (Morgaan, B., et al.).*

Figure 5.3-2 *The "Software Iceberg" (Grumman presentation on S.W. Engineering, 1983).*

Figure 5.3-2-.4 *Software and Hardware Cost Trends (Rachowitz, B., et al.).*

Figure 5.3-2-B *Hardware Capacity Effect on Software (Alford, M.W.,et al.).*

Figure 5.4-2 *Software Life Cycle Phases (Naval Electronic Systems Command).*

Figure 5.4-2-A *Two Software Life Cycle Approaches (adapted from Naval Electronic Systems Command...and Schindler).*

Figure 5.4-3 *Relationship of Management and Technical Software Tools to the Software Life Cycle (Morgan).*

Figure 5.4.4-1 *Relationship of Hardware and Software Life Cycle Activities (Naval Electronics Systems Command).*

Figure 5.4.4-2 *Life CycIe Comparison of a Hardware/assembly and a Software/Integration Project (adapted from Alford, M.W., et al., and Naval Electronic Systems Command).*

Figure 5.4.4-3 *Cost of Software Fixes at Different Time of Detection (Alford, M.W., et al.).*

CHAPTER 6 FIGURE SOURCES:

Figure 6.1 *Simplified Concept of Design/Production Integration (SP-4 CAD/CAM Integration Subpanel, 1981).*

Figure 6.1-A *Shipbuilding Design Production Integration (IBID).*

Figure 6.1-1 *Integration of CAD/CAM/MIS Islands" of Automation for Future Productivity Enhancements (excerpt from "Activity Roadmap" for SP-4 Software Tools Project: Progress Report No. 1; 21 June 1983- Sturgeon Bay, Wis.).*

Figure 6.2-3 *Conventional and ZOPM Shipbuilding (Chirillo, L.D., 1981).*

Figure 6.2.3-A *Pivotal Role of Design in Japanese Shipbuilding (adapted from Risenfeld, R.F.).*

Figure 6.2-7 *Pallet Concept for Zone Outfitting (Chirillo, L.D.).*

Figure 6.3-2 *Conventional and Integrated View of CAD/CAM (SP-4 "CAD/CAM Integration Subpanel'').*

CHAPTER 7 FIGURE SOURCES:

Figure 7.1 *Schematic of Product Work Breakdown Structure (PWBS) (Chirillo, L. D., 1983).*

Figure 7.2 *Product Work Breakdown Structure (PWBS) Interaction "with CAD/CAM (adapted from project notes).*

Figure 7.3 *Time Phased Introduction of Zone Outfitting to Shipbuilding (adapted from project notes).*

Figure 7.3-A *Relationship of CAD/CAM Integration via the SIP (adapted from project notes).*

CHAPTER 8 FIGURE SOURCES:

Figure 8.1 *Integrated CAD/CAM in a Shipyard of the Future (adapted from study notes; Albert M., Chen R., et al., Pouder, R., and Ramsey, R.).*

CHAPTER 9 FIGURE SOURCES:

Figure 9.1 *Allocation of Costs in a Typical Software Development Cycle (Naval Ocean Systems Center...).*

Figure 9.3.5 *Software Costing Categories and Factors (Bangs, A., et al.).*

Figure 9.7 *Cost Benefit Curve (Cotton, I.A.).*

CHAPTER 10 FIGURE SOURCES:

Figure 10.1 *Software Life Cycle and Classes of Software TooIs (ADA-STARS Program; DoD Presentation, 1983).*

Figure 1O.1-A *Life Cycle Phases with Specific Software Tools Assigned (prepared for report).*

Figure 1O.1-B *Role of Software Tools as a Foundation for CAD/CAM Integration (prepared for SP-4 Progress Report No. 1).*

Figure 1O.1-C *Software Synthesis and Software Development (prepared for report).*

Figure 1O.1-D *Software Development Process Phases (prepared for report).*

Figure 1O.1-E *Software Tasks and Functions in Development Phases (prepared for report).*

Figure 10.1-F *Tasks and Functions of Software Phases - Design (prepared for report).*

Figure 1O.I-G *Tasks and Functions of Software Phases - Development (prepared for report).*

Figure 10.1-H *Tasks and Functions of Software Phases - Integration (prepared for report).*

Figure 10.1-I *Tasks and Functions of Phases - Deployment (prepared for report).*

Figure 10.3 Classification of Software Tools by Feature (Houghton, R.C.).

APPENDIX G

BIBLIOGRAPHY

1. *Activity Roadmap for SP-4 Software Tools Project: SP-4 Software Tools Project: Progress Report #1; 21 June 1983; Sturgeon Bay, Wis.*

2. *AlAA Computer Systems Committee Software Tools Survey (Vol's 1 and 2-1980).*

   Rome Air Development Center, U.S. Air Force Systems Command, Griffiss Air Force Base, N.Y., pp. 1-315.

3. *Albert, M. "Fixturing Keeps FMS on Track". Modern Machine Shop (March 1984), pp. 50-59.*

4. *Alford, M. W. and J. T. Lawson, Software Requirements Engineering Methodology (Development). Report No. RADC - TR-79-168, Rome Air Development Center (ISIE), Griffis AFB, New" York (1979), pp. 4-6, 24-31, 70-75**

5. *Allan, R. "1982 Technology Forecast: Industrial Electronics". Electronic Design, (7 January 1982), pp. 94-118.*

6. *Analysis of Critical Issues in the U.S. Shipbuildkg Industry. A report to the National Center for Productivity and Qualil}y of Working-Life, by Pugh-Roberts Associates, Inc., (16 January 1978), pp. 1-28.*

7. *Anton, P., P.L. Arthur, and 3. W. Duncan, Technological Forecasting: Tools Techniques, Applications. Report #115. AMA Bulletin. New York, N.Y.: Research and Development Division (1967), pp. 7-9,24-26.*

8. *Bangs, A.; G. Foreman, J. Green, E. Rodriguez, R. Waina. Predictive Software Cost Model Study, Avionics Laboratory of U.S. Air Force, Wright Pattersen AFB, Ohio, June 1980, pp. 21-25, 36-45, 101-107.*

9. *Barkmeyer, E., C. McLean, and M. Mitchell. "A Computer Architecture for Small-Batch Manufacturing." National Bureau of Standards in IEE Spectrum* (May 1983), *pp. 59-64.*

10. *BeIady, L.A. and M.M. Lehman. "A Model of Large Program Development". IBM Systems Journal (No. 3-1976), pp. 225-252.*

11. *Blackburn, J.F., in European Scientific Notes (ESN 37-7; 31 July 1978) Office of Naval Research, London, from proceedings of the 3rd. Int. Conf. on Engineering Software Imperial College, London, England (April 1983), pp. 254-256.*

12. Borrell, J., "Solids .Modeling Becomes The Driving Force Behind Automation." Digital Design (Nov. 1983) pp. 88-89.

13. Bright, J.R., Technology Forecasting for Industry and Government: Methods and Applications. Englewood Cliffs, N.J., Prentice-Hall Inc. (1968), pp. 146-149; 153-157; 210-211; 270-275; 359; 430-435.

14. "CAD/CAM - The Factory Integrator". (No Author) - Production Engineering (April 1983). pp. 65-67.

15. Chen, R. and E. Cuthill. Computer Science Research and Development in Support of Ship Design Production and Repair (1978), Office of Naval Research Report by David W. Taylor Naval Ship Research and Development Center, Bethesda, MD., pp. 5, 8-11, 54-63,70-72.

16. Chirillo, L.D. SP-2 Outfitting and Production Aids, a presentation to the IREAPS Technical Symposium, Boston, Mass. (25 August 1983), pp. 1-8.

17. Chirillo, L.D. SP-2 Production Techniques, A presentation to the IREAPS Technical Symposium, Baltimore, MD. (15 September 1981), pp. 1-5.

18. Committee on Navy Shipbuilding Technology, Marine Board and Commission on Engineering and Technical Systems, NationaI Research Council. Productivity Improvements in U.S. Naval Shipbuilding National Academy Press, Washington, D.C. (1982), pp. 3, 44, 50, 51.

19. Cotton, I.A. "Methodologies for the Cost-Benefit Analysis of Computer Graphics Systems" (Based on NBS Technical Note 826 by I. Cotton). Computer and Graphics, Vol. 1, (1975) pp. 33-43.

20. Department of Defense (DoD) Directive. "Management of Computer Resource in Major Defense Systems. No. 5000.29 (April 26, 1976), pp. 1-2.

21. Deshler, D. and D. L. Winchester. "Automated Tools for The Analysis and Design of Red-Time Software." Reprint, 6 pps., Hughes Aircraft Co. (1983) pp. 1-6.

22. Diesslin, R. L., A Conceptual Information Model ... . for Outfit Planning IITRI Project - K06008; IIT Research Institute, Chicago, IL. (September 23, 1982), pp. 3-19,31-51, 57-66.

23. Diesslin, R. L., A Survey of CAD/CAM Technology Applications in the U.S. Shipbuilding Industry (1983), IITRI Project K06011; 11T Research Institute, Chicago, IL. (January 1984), pp. 2, 6, 109-112.

24. Doetsch, E.C., "CAM: Missing Link Between Management and Production Line. "Electronic Design, (7 January 1972), pp. 184.

25. Folkman, J. "DNC: What is it? Tooling and Production (Jan 1983), pp. 70-73..

26. *Fowler, E. B. "NAVSEA CAD/CAM Project Organization Alternatives:' Enclosure -33 pps. - Cover letter dated 20 April 1981 to Chairman Ship Productivity Committee (June 3-5, 1981 SP-4 Meeting, Atlanta, GA.), pp. 6-14.*

27. *General Electric Company; "Factory WITH a Future Products and Services:' Automation Planning Guide - A Step by Step Approach to Automation (1980 - G.E. Technical Brief).*

28. *Gitto, P. Economical and Viable CAD/CAM, DNC Systems for Large and Small Corporations,, Paper presented at 14th AMTC (March 13-16, 1977), pp. 1-18.*

29. Graves, Dr. R. J. and Dr. L. F. McGinnis, Jr., A Decision Support System for the Outfit Planning Problem: Modeling, and Conceptual Design. Report presented for Dept. of Commerce, No. 2 MA-RD-940-80055 (Feb. 1980), pp. 1-63.

30. *Grumman Data Systems. "Distributed Numerical Control" Corporate Presentation Materials (April 1982).*

31. *Hagan, S.R. An Air Force Guide for Monitoring and Reporting Software Development Status. No. AD-AO 16-488, Air Force Systems Command, Hanscom AFB, Bedford, Mass (1975), pp. 8-12,57-66,78-82.*

32. *Hart, J. L., " Technological Forecasting from Board Room to Drawing Board." Machine Design, (12 Feb. 1976), pp. 90-93.*

33. *Houghton, R. C., Jr., Software Development Tools (1982). National Bureau of Standards Special Publication 500-88, pp. 1-143.*

34. *Hymowitz, C. "Manufacturers Press Automating to Survive, but Results are Mixed". Wall Street Journal (March 11, 1983), pp. 1.*

35. *IBM Manufacturing Industry Marketing, " Computer-Aided Manufacturing Production Operations, Westchester Avenue, White Plains, NY (No Date),*

36.- *Inglesby, T., "Getting to the Heart of CAD/CAMJ' Assembly Engineering (June 1982), pp. 46-47.*

37. *Inglesby, T., "New CAD/CAM Entries have designs on the Future:" Infosysystems (Nov. 1982), pp. 56-62.*

38. *Keller, E. L., "Industrial Automation: Awaiting the Final Link - Standards:' Systems and Software (June 1984), pp. 96-97.*

39. *Krouse, 3. K., "CADI/AM - Bridging the GAP from Design (12 June 1980), pp. 117-121.*

40. *Krouse, 3. K., "CAD/CAM Broadens Its Appeal:' Machine Design (22 April 1982), pp. 54-49.*

41. Kull, D. To Raise Productivity Work Smarter, Not Harder." Computer Decisions (March 1984), pp. 165-189.

42. Langler, G.H. "It's Time For Computer Integrated Engineering" (Test and Measurement World (January, 1984), pp. 54-56.

43. Maritz, P. "Software Development: Mini-Micro Systems (December 1983), pp. 211-216

44. Marks, L., "CAD Systems Update" Electronics Packaging and Production (May 1982), pp. 55-70

45. Martino, 3. P., Technological Forecasting for Decision Making., New York, NY, American Elsevier Inc. (1972), pp. 65-69,287-295,331-343.

46. Morgan, B. And R. Skirkanich, "Large Scale Software Design Management Systems: Application Study and Implementation . . ." From A1AA Computer in Aerospace Conference, Los Angeles, Calif. (1979), pp. 1-7.

47. Morrison, L., "CAD: Snake oil or cure-all?' Plan and Print (May 1983), pp. 22-23.

48. Naval Electronic Systems Command., Computer Software Life Cycle Management Guide. NAVELEXINST 520023 (Computer Resources Management Guidebook Series For NAVELEX Project Managers - 1 March 1979, Sections/pp. I; 1.2, 1.3: II; 2.1, 3.0, 4.1.1, 5.1.2.

49. Naval Ocean Systems Center and Armed Forces Communications and Electronics Association, NOSC/AFCEA 3rd Intelligence Symposium Software Tools and Productivity (Proceedings), 16-18 November, 1982; pp. Session 2.

50. Nelson, R. L. "Increasing the Factors for CAD/CAM Growth." CAD/CAM Technology (Winter 1982), pp. 25-27.

51. Olenzak, A.T. "Technological Forecasting: A Pragmatic Approach:' Chemical Engineering Progress, Vol., 68, No. 6 (June 1972), pp. 27-32.

52. Peper, P., "Implementing CAD/CAM Manufacturing Operations (White 1984), pp. 26-27.

53. Perkins, D.R., P. G. Sorenson, J. P. Tremblay, and E. D. Wig. "A Survey of Some Automated Aids for Systems Analysis and Documentation:' INFOR, Vol. 19, No. 3 (August 1971), pp. 205-207, 127-218.

54. Pouder, R. and R. Skirkanich. Computer Automated Shipbuilding Technology. Grumman Data Systems Corporation overview report (24 April 1981), pp. 21-23,38-40.

55. Rachowitz B. and G. Sandier. Software Cost Estimating . . . A MANAGEMENT VIEW. Paper from Grumman/University Technology Forum (22-23 October 1979), pp. 1-14.

56. Raker, D.S., "CAD Angles . . .". Plan and Print (February 1974), pp. 6-7.

57. Ramsay, R., An Overview of the National Shipbuilding Industrial Base. paper presented at the 19th annual technical symposium of the Association of Scientists and Engineers of the Naval Sea Systems Command, Washington, D.C. (April 1982), pp. 11-26.

58. Ramsay, R. Approaches to Improving Shipbuilding Productivity, paper presented at the 20th anniversary technical symposium of the Association of Scientists and Engineers of the Naval Sea Systems Command, Washhgton, D. C. (March 1983), pp. 8-215.

59. Riesenfeld, R. F. Recommendation's for Computer Utilization in Shipbuilding. Dept of Computer Science, University of Utah. Report No. (ONR) 0001 477 C0157 (21 Nov. 1978), pp. 8-29.

60. Roegesgger, T. B. "Software Aspects of Factory Machine Control." Computer Design (April 21, 1983), pp. 131-137.

61. Schindler, M. "Software Automation Attracts the Programming Bottleneck." System and Software . . . Editorial Series Reprint for the systems integration re-printed from 12 Nov., 1981- Electronic Design, pp. 11-16, 3-35.

62. Singer, P., "Mid-Range CAD/CAM/CAE Systems Dominate Graphics Display Terminals:' Mini-micro systems (July 1984), pp. 223-228.

63. SP-4: "CAD/CAM Integration Sub-panel 3; Lina ChairmanJ' Notes from SP-4 sub-panel meetings, Atlanta, GA., (June 4-5 1981), pp. 1-7.

64. SP-4: "Meeting Notes:' From SP-4 Design/Production Integration Panel of National Shipbuilding Research Council meeting, Atlanta, GA (23-24 March 1982), pp. 3-5.

65. Sternberg, 3., "CAD/CAM - more than a drawing board . . . Less than a Miracle . . . A revolution in Industry:' Product Design & Development (July 1983), pp. 22-29.

66. Thayer, R. N. from Panel talk on Air Logistics Center, Calif. (A/AA Computers in Aerospace Conference, Los Angeles, Calif. (1979).

67. Thibodeau, R., An Evolution of Software Cost Estimating Models Rome Air Development Center, Griffiss AFB, N.Y. - Report No. CR-1-940 (June 1981), pp. 14-20.

68. Tuck, 3., "Successor to CAD/CAMJ' Circuits Manufacturing (February 1984), pp.16. '

69. "Use of Numerically Controlled Machine Tools Aboard U.S. Navy Destroyer Tenders," Defense Technical Information Center, DLA, Cameron Sta., Alexandria, VA (February 1970), pp.3-10, 21-28.

70. *Ware, R. B., "Systemics: What a systematic needs to know and do' Infosystems (May 1978), pp. 106.*

71. *Wiedenhaefer, P. "The Engineering/production Integration Process: Graphics, Interactive computing and data base." In proceedings of the shipbuilding design/production integration workshop - Vol. I, Atlanta, Ga. (January 18-21 1981), pp. 16-21.*

72. *Zdeblick, W. J., "Process Design by Computer:' IEEE Spectrum (May 1983), pp. 56-58.*