UCRL-95441
PREPRINT

Current and Future Issues in the Design, Analysis, and Implementation of Distributed Computer-Based Control Systems

Gregory J. Suski
(LLNL)
and
Michael G. Rodd
University of Swansea

Seventh IFAC Workshop on Distributed Computer Control Systems
Bonn, Germany
September 29-October 3, 1986

September 1986

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

Lawrence Livermore National Laboratory

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government thereof, and shall not be used for advertising or product endorsement purposes.

Current and Future Issues in the Design, Analysis, and Implementation of Distributed Computer-Based Control Systems*

Gregory J. Suski
Lawrence Livermore National Laboratory/University of California
Livermore, Ca., U.S.A.

Michael G. Rodd
University of Swansea
Swansea, Wales, U.K.

Abstract

The field of distributed computer control system engineering as applied specifically to laboratory, manufacturing, and process control applications, encompasses a spectrum of diverse computer disciplines. Over the past several years, researchers and applications engineers in this area have focused their attention on a variety of issues. The issues deemed most important have varied over time according to developments in hardware and software technology, trends towards more stringent performance and reliability criteria, evolution of internationally accepted standards, and the emergence of new application areas.

Successive workshops in the IFAC series on Distributed Computer Control Systems (DCCS) have provided forums for discussing these issues. Early workshops concentrated on systems architectures and relevant issues such as programming languages, open versus closed architectures, and the applicability of emerging standards. Most recently, discussions have intensified on control system issues pertaining to flexible manufacturing systems and the development of modeling techniques to allow verification of system designs. It is evident that the emphasis changes with time. It has also become apparent that the evolution of technology frequently leads to the revisiting of prior issues for which new improved solutions become available.

This paper summarizes current important issues at present in distributed computer control systems. Starting with the papers and discussions of the Sixth IFAC workshop in Distributed Computer Control Systems as a base, the recurring issues of DCCS are reviewed and topical issues of interest are described. The paper concludes with a discussion of probable future interest areas in distributed computer control systems.

Introduction

The field of distributed computer-based control system (DCCS) engineering is nearly a decade and a half old. Its birth corresponds more or less to the introduction of low cost minicomputers in the early 1970's. Since the first distributed systems were introduced, their use has become pervasive in all areas of process control. Applications areas include manufacturing, large research experiments, utilities, and raw material production. From the perspective of process engineers, the growth of this engineering discipline has been important in addressing requirements for better reliability, process reproducibility, lower costs, improved productivity, enhanced flexibility, and increased visibility into process operations. From the computer and control engineers' viewpoints, the field has proven to be fertile ground for the application of the newest technologies emerging from the industrial, information processing, data communications and office automation sectors of the electronics industry.

The DCCS community is continually incorporating new technologies and revisiting prior design strategies. This fact is clearly documented in technical meetings such as the 1985 IFAC Workshop on Distributed Computer Control Systems.[39] Researchers and developers are concentrating on the development of improved design and analysis strategies aimed at reliable, improved performance, cost-effective solutions to application requirements.

To the users and others who must pay the cost of control systems development, some of this effort appears unwarranted and counter-productive. This perception is often reinforced when engineering peers in this field disagree, sometimes fundamentally, as to optimum DCCS design strategies.

*Work partially performed under the auspices of the U.S. DOE by the Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-38.

Cost conscious managers of projects requiring process control systems often question the value of embarking on new DCCS design efforts when systems designed for other projects seem to perform adequately. Such inertia in moving towards new technologies is diminishing in certain application areas such as automation. There it is recognized that fundamental improvements in process control performance and functionality are required in order for the process itself to produce competitive products.

The purpose of this paper is to survey the factors which lead to continued evolution in DCCS methodologies. An introductory section provides background summarizing the current role and characteristics of such systems. This is followed with a summary of the technical issues relevant to DCCS design, a review of the major technical areas which must be addressed in DCCS designs, and a discussion of influential factors in the DCCS development environment. The remainder of this paper will concentrate on the relevant emerging technologies and issues which are of interest to researchers and designers of new systems. Included are speculations as to their possible impact on next generation distributed computer-based control systems. A summary provides an overall assessment of major DCCS trends.

Evolution Toward Distributed Computer Control Systems

Distributed computer control systems are applied today to small and large processes in many dissimilar fields. The essential functions of any DCCS are to allow predictable control of an ordinarily non-deterministic process and to acquire data concerning that process. The DCCS applications domain includes any collection of physically independent tasks which are capable of being electronically controlled and monitored, and which must function together as a system to accomplish an objective. These include research facilities such as accelerators, fusion research machines, and biomedical laboratories; manufacturing systems; power production plants; agricultural installations; mining operations; etc.

Historically the most prominent early uses of computer control were in the chemical process control industry beginning in the late 1950's.[2,26,38] These early applications were at installations operated by Texaco (aided by TRW), American Oil, and Standard Oil.

In these first uses of digital computer control, a single computer such as an IBM 1700 series machine was used in a supervisory role to calculate and in some cases control set points on analog controllers.[2] The analog controllers remained responsible for the primary control functions. The data available from sensors oriented towards prior control schemes limited the information available for calculations. Digital computers were typically not in direct on-line control of processes due in part to the poor reliability of the computers. A major reason for introducing computers in such systems was the need to improve the yield of processes through more timely adjustment of control valves and switches in response to changing process conditions and product specifications. Early computer control systems suffered from high cost and poor performance in addition to poor reliability. The cost of such systems was justified by allocating to them many sensors, actuators and functions.[2,40] While such factors limited the number of applications areas, the presence of machine intelligence in the control system justified their use. These early successes provided the impetus for the later development of lower cost systems better suited to process control.

In early 1962, the first uses of digital computers for direct process control appeared in which analog control electronics were bypassed entirely. A key demonstration of the technology was accomplished by Imperial Chemical Industries, Ltd. in England using a Ferranti Argus computer. It was now feasible to add interaction easily between control loops and, during this period, the first digital control oriented programming languages were developed. Progress remained limited by the availability of inexpensive, reliable computers. The development of the low-cost minicomputer was to change that situation.

Multicomputer Control Systems

As computers became smaller and less expensive during the late 1960's, the applications domain expanded to include progressively smaller processes. Inexpensive minicomputer-based control systems were made possible through the development of low-cost solid-state electronics and particularly integrated circuit technology. This led to the economic feasibility of using digital computer control in simpler processes.

More importantly, this development made it possible to move the computer into educational institutions and their laboratories. By the early 1970's, a growing pool of graduating engineers with basic skills applicable to digital computer-based process control was entering the industrial and scientific communities.

It is estimated that in the period from 1970 to 1975, there was an order of magnitude increase (to approximately 50,000) in the number of process computers in use. As microcomputer technology became established, the cost and size of process control computers decreased while their reliability improved substantially. The development of more sophisticated languages and operating systems for minicomputers hastened their acceptance, particularly by the end-users who used the flexibility introduced by higher level software to perform more sophisticated analyses and control of processes. The development of minicomputer networks in the latter 1960's and early 1970's provided the final piece needed for the emergence of distributed computer control systems.

This transition from first to second generation distributed control systems is characterized by their hierarchical nature (Figure 1). Such systems were marketed commercially starting in 1975 with a product introduced by Honeywell. In the second generation, the concept of local versus central computer control emerged. Smaller computers were located near the devices they controlled or

monitored, often providing a local control panel with basic functionality for maintenance purposes. They were interconnected, usually via serial RS-232C communication links at speeds up to 9600 bits/ second, to one or more central computers which generally had more processing capability, more memory and disc storage, peripherals such as printers, and a high functionality operator interface.

These systems used graphics to display system status, supported operator commands entered through keyboards, and often included graphical input devices such as light pens. The central computers performed analysis in addition to control and also implemented feedback loops that spanned across interfaces connected to different local computers. Frequently the size of the process, or requirements on functionality, led to additional levels of more powerful computers. These in turn were connected with point-to-point network connections to the middle level computers. With multiple levels of computers in such a system, a hierarchy of computers and functions emerged. Accordingly, these became known as hierarchical distributed computer control systems.

The majority of the distributed computer control systems in use today employ second generation architectures. Figure 2 illustrates a model of these systems and the environment in which they must operate.

At the top of this model appears the overall organizational structure. It is simplified here to show a layer of top management which is responsible for the overall strategy of operations. If this is a manufacturing organization, this level may be responsible for decisions as to the type and quantity of items to produce. This level deals with long term decision making and can employ computer-based information, resource allocation, and decision systems of varying complexity.

The next level manages the daily operations of the facility which may have more than one process active. Management information system (MIS) aids may also be used at the level. Process status information in the MIS may require frequent updates to allow effective process supervision and reporting. The process appears in Figure 2 at the lowest level. It is characterized by a set of inputs (resources) and outputs (products) and is monitored and controlled through a set of sensors, actuators, and control elements. The process is controlled and monitored by one or more operators according to requirements established at the level of daily operations management.

Requirements on decision responsiveness increase in going from the top management level down to the process level. At the upper levels, decision time frames of days, weeks, and even years are typical. Conversely, the process may require responses in the minutes, seconds, milliseconds, and submillisecond ranges. The connection between management and the process is primarily through the DCCS.

To accommodate a range of requirements, the DCCS is generally hierarchical in its distribution of functionality. Electronic interfaces are used to allow connection of distributed small computers (or controllers) to sensors and control elements. The distributed control computers provide local control functions and often a local control station. They may process local feedback loops. They must be capable of handling information transfer not only with the process devices but also with one or more other computers in the DCCS. Computer to computer information transfers include command, status, and often synchronization information.

Other computers in the DCCS are responsible for higher level functions including supervision, coordination of processes, feedback loops spanning across distributed processors, and computationally intensive tasks. Computers at this level are also responsible for performing centralized control tasks and for providing the primary operator interfaces.

The characteristics of the communications media between devices and computers, and the degree to which process data can be transferred directly between management information systems and the process control system, are important issues. The latter is particularly important in the manufacturing community. These issues will be covered later during the discussion on characteristics of third generation architectures.

There are several reasons why distributed computer architectures are well suited to process control. For large processes, they allows successive decomposition of the process control problem into manageable elements. One of the principal problems with single computer control of large processes is the complexity of functions which must be co-resident in the central computer. The need to handle real time process control activity in the same computer that performs analysis and computation represents a performance bottleneck. It also presents a significant software development problem, requiring much greater development control and coordination, and more difficult testing of the software.

There are economic advantages in DCCS hardware costs. During the period of their initial acceptance, there was the additional advantage that a relatively low cost investment could result in a proof-of-principle prototype. Of even greater significance, it is generally less costly to distribute the intelligence in smaller computers than to centralize intelligence unnecessarily in higher performance computers.

Another major benefit of a distributed system is its natural characteristic of fault isolation. A failure in one portion of a distributed control system can normally allow the remainder of the process to continue at some level. In addition, the operational portion of the system can assist in reporting and diagnosing the fault. Other significant benefits include the availability of distributed control stations (operability) and the ability to add control functionality without major system impact (extensibility).

Essential Aspects Distributed Computer-Based Control System Design

DCCS development is characterized by traditional engineering practice. For instance, the major elements of DCCS development include specification, design, optimization, implementation, testing, and maintenance (including extension). These elements are strongly interdependent. For example, selection of the hardware and software technology to be used in a DCCS has a strong impact on the corresponding methods of its implementation, testing, and maintenance.

DCCS applications begin with a preliminary definition of the system criteria, including the technical requirements, cost, and schedule. Technical requirements encompass safety, functionality, performance, reliability, operability and extensibility. Requirements definition is often an iterative procedure in which the design of the process is itself influenced by the capabilities of the control system. For a large, long term project, much of the control system functionality may not be precisely defined. Because extensions can be accommodated without fundamental changes to the remainder of the system, DCCS technology is well suited to such applications.

The performance of the control system is usually a major criteria. In process control systems, the rate at which the system is sampled and controlled should be at least twice the rate at which significant changes can occur (Nyquist criterion). In this paper, a system which functions at that rate will be said to be operating in real time. In order to assure that a given performance level will be achieved, control systems are frequently simulated or modeled during the design process.

Another key criteria is reliability. This includes a wide range of issues including the fault tolerance of the system[32] (e.g., degree of fault isolation, response to software errors), its long term maintainability, method of fault recovery (e.g., partial versus total system restart), and the need for system diagnostics for on or off-line operation.

In the absence of additional constraints, DCCS designers must select from an enormous array of currently available technology options. These options are categorized here into eight groups (Figure 3).

Communications

The collection of hardware and software which comprise the communications scheme between the various electronics systems forms the backbone of any distributed computer control system. It has a fundamental impact on all other design issues and the ultimate ability of the system to meet its criteria. Selection of the communications system is also critical since it is one of the most difficult DCCS elements to replace or functionally extend once it is in use.

There has been increasing acceptance in recent years of the seven level Open Systems Interconnect (OSI) reference model (Figure 4) for interprocessor communications developed by the International Organization for Standardization (ISO) in Geneva.[42] This model has provided a framework for the development of communications standards. These can be implemented in either hardware or software, depending on their complexity. An important contribution of this model is the establishment of a universally understood terminology for describing communications functions.

The DCCS designer must choose not only a good communications system, but whether to choose one conforming to released standards, proposed standards, or perhaps defacto standards such as commercial LAN's from the office automation industry. The choice will affect the final cost and long term extensibility of the system. The designer must also choose whether to use separate systems for communication at low levels (e.g., to device controllers) and high levels (between computers).

The method of process synchronization must be addressed. Options include synchronization with data transfers on a common communications system, using an exterior clocking scheme, or using data with time stamps.[30]

The designer must choose the medium for data transfer (e.g., wire or fiber optics), whether data is transferred serially or in parallel, and whether communications system access is by demand, controlled by a master, or given via token passing. In an issue related to architecture, a fundamental decision as to bus versus point-to-point communications is required. The use of separate versus a common communications scheme for devices and processors is also important.

Deterministic communications system performance is often required to ensure sufficient communications access to meet real time requirements.[30] Therefore, a choice as to the type of protocol to be used is important. For example, the ubiquitous positive acknowledge and retransmit protocol (PAR) guarantees data transfer but has high overhead. Alternative protocols may give no assurance as to data transfer. Determination of performance bounds on networks can involve simulation, protocol modeling, and/or protocol verification. The latter techniques are not universally available.

Architecture

The architecture of a DCCS is based upon the communications system characteristics and the required process control functionality. Decisions as to whether a multilevel or a uni-level system are to be used depend more on the characteristics of the communications system than the process. However, the topology of interconnection among the hierarchy of computers will greatly affect functionality. Network topologies can include node-to-node, ring, bus oriented, star, hypercube, etc. The issues of which functions are to be loosely versus tightly coupled impact performance and fault isolation characteristics. Another architectural consideration

is the aspect of how real time state information is acquired. Are devices polled, are they accessed through interrupts, or are they represented by state information which can be accessed on demand?

Process Interface

The nature of the process interface must be understood and specified. The adoption of a standard can reduce the software and hardware development cost substantially, but may restrict the availability of control elements and sensors to the process system designer. Device characteristics and interface requirements must be identified in advance as much as possible. For example, a device which generates a need for attention every few milliseconds can seriously degrade the performance of certain interrupt driven architectures.

Function Allocation

The allocation of functions within the control system directly impacts its performance and often its fault tolerance. Within a multiprocessor system, handling of a function may be centralized or distributed. The allocation of functions can be done statically during implementation, pseudo-dynamically during system startup, or dynamically during system operation according to process activity. True dynamic allocation is much more difficult to implement, but has the advantage of allowing more optimal use of system resources. Simulation and complex analysis of statically allocated systems is often required to ensure meeting performance criteria.

Hardware

Major hardware decisions are made during the communications and architectural design of the system. However, within any given architecture, several hardware decisions must still be addressed. There is the issue of whether to standardize on a particular computer manufacturer or interface standard. This may involve decisions balancing the cost of long term system maintenance against flexibility in optimizing performance and functionality. As important as the decision to standardize is at what level to standardize. Open architectures which specify performance levels but not method of achieving them are often preferred over closed architectures at the initial specification phase of large projects. Selection of a closed architecture is may then be accomplished before the final system design.

Software

Software for DCCS applications can be complex and is an increasingly greater portion of the cost of such systems. Initially, the DCCS designer must choose programming languages, operating systems, data base managers, and communications packages. Where standards based hardware subsystems have been adopted, this task is often eased by the corresponding availability of standard software. However, in general, software for specific applications must still be developed, whether through traditional programming techniques or more advanced program writing systems. A specific software project management methodology must be identified and put in place. Quality assurance of software through the specification, design, implementation, testing and maintenance phases is a recognized necessity today in the development of any large system. The DCCS designer generally must also accommodate requests for high level functionality which may be provided by pre-packaged software such as decision support systems and resource scheduling systems.

Data

The organization of data within a distributed computer control system is second in importance only to selection of the communications system. Designers must identify the types of data required (e.g., state information, command logs, device setpoints) in each node of the system, how and when this data is acquired, its structure, whether it is of local or global interest, and what type of access controls are required. As the size of distributed systems increases and performance criteria are raised, decisions as to what data should be in central versus distributed data bases are required. When data must appear in more than one distributed location concurrently, issues of data consistency (i.e., all data instances having same value) and validity (data reflects current system state and is useful) must be addressed. As with most other areas of DCCS, there is no currently identified "preferred" scheme for dealing with this issue, although schemes using data tagged with time and age information appear viable.[30] Journaling of significant events, transactions, and commands may also be required for maintenance, safety, or legal reasons.

Human Interface

The most important contact operations personnel will have with many processes is through the process control system. Safety and ergonomic factors predominate in the design of the primary operator interface. Often taking the form of a control "console", the ability of this device to communicate quickly and clearly with an operator is essential to good system operability, both real and perceived. Where a console or control panel is used infrequently (e.g., for maintenance), it is important that it provide the minimum required functionality and that its design does not lead to unnecessary operator errors. Where an operator interface is used more continuously, it is important that it be a comfortable work station that does not lead to unnecessary operator strain.

Information displays, often based on color graphics systems, must be clear and properly oriented to minimize fatigue. Important data must be immediately accessible and the method of operator input should discourage improper system commands. Command entry must be simple and cross-checked by the system for validity. Often the control system designer involves operations personnel in the design of the operator console system.

Given the number of technical options to consider in designing a DCCS, it is not surprising that there is infrequent agreement on what constitutes an optimal system design. However, several "real world" factors represent strong influences on the final design of a DCCS. These factors are not necessarily technical in nature and are frequently specific to the environment in which the DCCS is being developed.

Relevant Standards

The availability of hardware or software conforming to established, proposed, or defacto standards (see above) can have a strong influence on the selection of elements in a DCCS. Well established standards generally have the most associated products, but are often based on aging technology. New and proposed standards offer newer technology, but with limited initial availability. In addition, industry acceptance of such standards is sometimes at doubt. Nevertheless, standards are a strong influence and can significantly affect final system designs.

Available Resources

The developer of a DCCS system is often faced with little time or insufficient budget to assemble new resources such as people, hardware development aids, software development systems and management tools. The use of currently available hardware and software for development will clearly influence the selection of such components in the final system. The lack of large project management expertise and aids can imply a practical limit on the scope of the development effort or impact the method of implementation (e.g., outside contracts versus in-house development).

Inherited Technology

The availability of suitable hardware from a previous process control system in the presence of strong DCCS cost restraints can reasonably be expected to impact the final system design. The availability of reusable software, such as functional libraries or compilers can also affect design decisions. Where the DCCS development effort must incorporate an existing process control system, hardware and software compatibility may be required. Alternatively, a "gateway" can be developed to marry the newer technology system with the older.

Expertise and Interest of Staff

The skill-base of the technical staff responsible for designing and implementing a DCCS can have strong impact on the final design. While most people are eager to acquire new technology skills, they also develop areas of personal interest and competence. People are naturally more productive and satisfied when they are working in an area of interest. This "fun factor" aspect ordinarily has a major influence on design discussions.

Organizational Biases

Organizations have biases as do people. Often the use of a particular standard will be directed due to a corporate commitment or a particular manager's bias. The use of a competitor's products or standards may not be allowed. A history of poor relations with a particular vendor may preclude future use of that vendor's products. Conversely, successful past interactions may lead to sole-source contracts with favored suppliers. These factors work to constrain the options available to DCCS engineers.

Design Process

As systems become more complex, reliability and safety requirements increase, and development schedules get tighter, there is a growing emphasis on using methodical, well documented design processes. The results of such formal design procedures are often different from more ad hoc approaches. Designs which are contracted to third parties may differ from those performed in-house. Review committees may also alter final designs. There are many decisions leading to a final DCCS design. Any factors influencing these decisions will impact the result.

Validation

Requirements on the engineer to prove that a system design will perform the tasks intended can influence the technologies to be used as well as the method of design and implementation. Currently, both modeling and simulation of distributed control systems are primarily used to assure the validity of the design. Prototyping essential parts of the system is also frequently used to verify a conceptual design. This latter approach maintains staff interest during the development phase, allows people to become familiar with the technology, and provides data which can be used to verify or revise the final DCCS design.

We have summarized the reasons for the growth of distributed computer control systems, addressed many of the relevant DCCS technologies, and discussed the non-technical issues which strongly influence DCCS designs. The reader is familiar by now with the scope of the DCCS design problem. We will next look at the factors which are emerging today which are likely to cause further evolution in the design of these systems.

Emerging Trends and New Directions in DCCS Development

DCCS designs have undergone rapid evolution since such systems first appeared. What are the reasons behind this? Is it simply that, as engineers, we are so attracted by new technology that we may automatically seek to incorporate it without valid justification? The answers are found in three principal areas.

First, the simple availability of new technology actually is a driving factor, but not just for its interest value. In designing systems for long term operation and extensibility, new technology, particularly that based on standards, is more likely to be available throughout the life of the system to aid in maintaining and extending it (Figure 5). In addition, such technology is generally available at

lower cost as older technology becomes obsolete and scarce. It is easier to recruit high quality engineering talent if system designs are based on non-obsolete technology. Reducing DCCS costs motivates use of new technology with improved cost/performance ratios. However, new DCCS designs often exceed the cost of their predecessors. This leads to another area of influence on new DCCS designs.

The second major factor in DCCS evolution relates to process related requirements for new functionality. Pressures are increasing for improved productivity, reduced waste, lower operational manpower requirements and greater reliability. To meet these criteria, DCCS designers are required to put more functionality into the control system. Readily available standard hardware and software for common functions may be adopted in order to provide the time, budget, and resources needed to develop new functionality. In many cases, the need for more rapid response to changing process conditions leads to higher performance requirements on DCCS implementations. This motivates the use of higher power computers and improved communications architectures.

The third major factor affecting new DCCS designs results from issues related to economics, education, sociology, ethics, and political influences.

New DCCS Technologies Impacting Performance and Architectures

The remainder of this article will discuss these three factors - emerging technologies, new applications criteria, and emerging issues. The technologies which will have impact principally on the system architectures, performance, and applications domain will be discussed first. Examples of new areas of application and requirements will be described to put these technologies in perspective. Next will follow a discussion of those additional technologies which are being developed to meet new requirements on functionality. A discussion of emerging non-technical issues will conclude this section.

Communications

Current and emerging developments in DCCS communications will have major impact on the systems of the future. The most significant of these developments is the emergence of bus oriented interconnection techniques in which all computers and, in many cases, the process interfaces themselves, may use the same physical communications medium. This implies that any computer can access any other computer or device in the system directly, without going through an intermediate processor or series of processors.[27] The system performance is improved by eliminating routing delays and allowing greater distribution of functions among processors. Extensibility is also improved, since adding nodes or additional computational power is simply accomplished through additional connections to the common bus.

Actual implementations of such communications schemes may employ high speed serial communications media utilizing fiber optics[4] or wires. Access control to such media becomes a critical design issue since it affects the deterministic behavior of the communications system. Standards work in this area is currently of great interest.

One of the more visible efforts is the General Motors Corporation inspired MAP specification which utilizes a token passing technique to control access to a broadband, 10 Mbits/second communications link. The broadband nature of the physical medium allows more than one such link to exist on a single cable. There is also a joint PROWAY/MAP effort underway studying variations to MAP to treat real time concerns. This effort may also resolve one of the major issues in MAP, its cost per node, which make it a formidable undertaking for smaller manufacturers. Such specifications are leading to open systems architectures at the device level which will provide universal interfaces between process control elements (e.g., actuators, robots, sensors) and computers. Version 3 of the MAP specification is expected in 1987. It is expected to also incorporate the TOP (Boeing Technical Office Protocol) specification to treat interconnection with office communications systems (Figure 6).[1,35]

MAP specifications are still evolving, and broadly based hardware support for it is estimated to be approximately one year in the future. However, the interim use of programmable gateways to provide a uniform interface to different manufacturers' devices is a commercially available viable approach today (e.g., Digital Equipment Corporation's Baseway).

MAP related protocols and standards will ease the task of interconnecting to different manufacturers' products. MAP suggests a system architecture in which all nodes use a common communications bus. While bus bandwidth considerations must still be addressed, such architectures are a clear trend in today's systems. They are characteristic of third generation computer control systems (Figure 7). Another major characteristic of third generation systems is their interconnection into the management information systems (MIS) which are used by operations and higher level management.

Future standards development in the communications area is generally being accomplished within the framework established by the OSI reference model. Relevant standards work will be targeted at meeting the needs of distributed data base architectures, stringent real time process requirements, higher speed communications media, and expanding requirements of interconnectivity between management information systems (MIS) and process control systems. There are now commercially available, fiber-optic based digital communications media operating at speeds in excess of 200 Mbits/second. Such media allow unprecedented opportunities for the sharing of real time data and coordination of distributed process control functions. The growing needs for interconnectivity between not only processes but facilities is leading

to global networks used for information, status, and scheduling. Interconnection of DCCS' to such networks can be expected as needs for closer production control increase. The Integrated Services Data Network (ISDN) work is particularly relevant to future development of global communications.

Technology is moving to eliminate point-to-point interconnection schemes and to provide high speed deterministic data communications. The traditional approach of selecting the DCCS information structure according to the physical nature of the plant will no longer predominate. The ease of information sharing will lead to information structures which are best suited to the characteristics of the DCCS.[27]

Until recent years, the technology applicable to the DCCS environment was derived principally from the office automation and information systems sectors of the computer industry. This was due to the greater economic incentives in providing products for their larger customer bases. Low cost DCCS systems continue to benefit from such development. The defacto standards represented by local area networks (LAN's), developed principally for office communications systems, are now being incorporated into new DCCS designs.

A more recent technology is the development of wireless communications systems to ease the installation of office automation products within a local area. DCCS designers should consider the impact on their design if physical cabling is not required. For instance, what are the possibilities presented by portable control nodes? The result of this development activity on DCCS engineering will be a decreased emphasis on developing communications systems and an increased emphasis on their use.

Computer Systems Technology

The revolutionary development of computer systems technology continues to have strong impact on the levels of performance and types of functionality which can be economically incorporated into new DCCS designs. Cost/performance ratios for processors improve two orders of magnitude each decade. Super microcomputers and highly parallel architectures are now becoming commercially available. In the near term, such processors can be easily added into third generation architectures to perform analysis functions traditionally relegated to CRAY class supercomputers.

Interconnection of DCCS nodes. In the longer term, the techniques of interconnectivity, synchronization, and data sharing developed for massively parallel architectures (such as hypercube, torus, BBN's Butterfly, Inmos' Transputer, etc.) will influence designs for similar functions in DCCS implementations (Figure 8). A good example of highly parallel, commercially available computers is the 65,536 processor Connection Machine from Thinking Machines, Inc. This highly parallel, interconnected machine has a performance factor of 1,000 Mips.

Simulation and Modeling. As supercomputer performance becomes more accessible to the DCCS designers,[11,36] thorough modeling and simulation of complex DCCS designs will become standard practice. A corresponding availability of very large capacity, inexpensive solid-state memory and online mass storage systems will allow important real-time control and status information to be instantly accessible. These factors will impact the nature of the DCCS design process, system performance, reliability, and the quality of operator interactions.

Personal Computers. At the other end of the spectrum, low level control functions are now being economically incorporated into personal computer (PC) based systems. The enormous competitive cost pressures on PC's and the growing number of interfaces and software packages available for such systems will lead to the increased use of PC's in future DCCS implementations. Used in processes with bounded, low level performance requirements, PC's will fill the need for low cost DCCS computer technology which over the last decade has been met by minicomputers.

Low-Cost Intelligence. There are now high speed array and image processors which can be attached to PC's and other small computers found in DCCS implementations. The price for these systems has dropped more than an order of magnitude during the last decade. New sensors required for process control often produced strings of data at high information rates. Such sensors may produce several thousand points of information per sample. Many sensors are now CCD or CID based imaging arrays producing on the order of one hundred thousand points per reading. In second generation hierarchical DCCS', such data was often handled with a limited number of expensive, higher level processors. This caused inherent delays in data acquisition and command processing. In third generation systems, the trend is to put more intelligence closer to the process, as hardware becomes less expensive and real time data sharing becomes more feasible.

This increase in computational power available to the DCCS designer will lead to the increased use of adaptive control techniques in the future.[24] Such techniques often require the use of system models to modify the operational parameters of a process, but their use has been restricted due to the computational complexity of the tasks involved.

CMOS Microprocessors. Finally, the availability of CMOS versions of standard microcomputer allows a new degree of intelligence distribution in process systems operating in harsh environments. With their low power consumption and dissipation, CMOS microcomputers can be packaged in sealed cabinets with built in battery backup. This allows DCCS node placement in environmentally difficult and geographically remote locations. The result will be an increase in the types of applications suitable for DCCS implementations.

While CMOS circuits are currently limited to clock rates less than 20 MHz, circuits based on other technologies exist or are under development. For instance, Gallium Arsenide technology holds long term promise of 100 MHz clock rates and the ability to be used in radiation environments.

Component Technology

In addition to developments in computer systems technology, the development of related hardware components will affect the design, functionality, and design methodology of DCCS systems in the future.

Sensors. The development of new sensor technology, particularly that associated with robot based automation, is a compelling force on DCCS technology. It is estimated, for instance, that the machine vision commercial market will exceed that of robotics itself by 1996. It has in fact doubled each of the last two years and can be expected to do so for at least the next three years.[21]

The flexibility of robot manipulators to handle objects of different characteristics is strongly influenced by its sensors. The contact switch is being replaced by tactile sensors, proximity detectors, and vision systems.[5,19,33] As the performance demands related to such sensors increase, improved DCCS computational power and responsiveness will be required. The result represents a long term trend in DCCS designs; machine intelligence is placed successively closer to the sensors and controlled devices (Figure 9). Local processing of video data has already been discussed. A logical trend is to see sensors in the future with sufficient embedded intelligence to performed programmed analyses and reduction of their own data. In addition, such sensors will have integrated communication functions for interconnection at higher protocol levels with the DCCS.

Custom VLSI. Maturing gate array technology will allow inexpensive chip implementations of common DCCS logic circuits. For instance, such technology will lead to a substantial reduction in the per-node costs of custom communications systems. It is now commonplace to have small quantities of arrays with up to 5000 gates fabricated by silicon "foundries" in order to replace circuit boards populated with LSI technology. Components with 10000 gates are becoming available, and development of a 500,000 transistor array is currently under development. This technology is growing (IC's are doubling in complexity approximately every 2-1/2 years) but is available now to interested designers.

In addition, access to gate array and custom integrated circuit technology is simplified by the concept of silicon compilation.[7] Currently in its infancy, silicon compiler technology provides schematic capture schemes to accept circuit designs, then performs chip layout and routing of interconnects according to rules appropriate to the technology and foundry being used. Mobile silicon foundries exist which allow local access to a facility for quick iteration of designs. Two day turnaround times from the completion of schematic capture to final chip are feasible. There is a continuing goal to reduce this time to a few hours or less. There is also continued development of silicon compilers to include simulation of circuits and provide design feedback. Several systems already exist which can predict the performance level of a design before final implementation.

Such technology will substantially lower replicated circuit costs while shrinking their size to the chip level. This implies inexpensive communications system connections, low cost interfaces to the process, and makes feasible special purpose computational electronics located near sensors.

The overall impact of advances in computer and device technology will be towards robots and process elements with semi-intelligent, intercommunicating nodes forming electronic nervous systems with which feedback and control will be performed as part of the robot or process itself. The programming of these systems will be highly complex and computer assisted.

Optical Disks. Another new technology which will impact DCCS architectures and functionality is that of the digital optical disk.[23] Optical disk technology is now commercially available in 5 1/4 and 12 inch formats. Optical disks have approximately 20 times the capacity of equivalent-sized magnetic disks (Figure 10). The 5 inch units are able to store over 500 Mbytes of data while the 12 inch units exceed one gigabyte of storage. This is an enormous quantity of information. Over 4 million standard typewritten pages can be stored on a single 12 inch disk. The technology is currently available in read only (CD-ROM) and write-once, read many (WORM) versions.

The principal advantages of optical disks are their density and the integrity of their data. High density disks are increasing in importance as DCCS systems become more complex. For instance, since they can combine large amounts of video, text and process data, they can be used to serve as an effective teaching aid for operators or as a diagnostic reference aid. These aids will be even more effective since motion sequences or animation can be included. In another application, the WORM technology can be used to journal system commands and events. The write once nature of the disk is an advantage since data becomes difficult to lose or alter.

The use of optical disk technology for the recording of digital information will partially reverse a recent DCCS trend towards less use of mass storage on lower level computers. These disks can be used to inexpensively archive process state data in distributed locations in order to aid in fault diagnosis and process characterization. In addition, because optical disks are much less susceptible to radiation than their magnetic counterparts, they will prove valuable in providing permanent logs of process history in environments where radiation may occur.

Multiple-write, multiple-read versions of optical disks are currently under development. Their availability will help fill the needs of future DCCS systems that will be processing and recording large amounts of process specific video data.

New DCCS Applications Lead to New Criteria

Scientific Research. The combination of technology availability and new applications requirements on distributed systems for control often make compelling arguments for adopting the latest technology. In the laboratory research environment, large scientific machines such as accelerators and fusion research facilities are becoming increasingly expensive. Accordingly, fewer such systems are being built. Those that are built must be controlled by highly reliable systems. These must maximize the experimental capacity of the machines of maximize benefit to the scientific user community. In such systems, the control functions are substantial, but are frequently overshadowed by the quantity of data that must be acquired, managed, and analyzed.

Computer Integrated Manufacturing. A more pervasive application forcing the use of advanced DCCS technology is Computer Integrated Manufacturing (CIM).[12,20] The manufacturing industry worldwide is faced with the challenge of transitioning to modern manufacturing methods in order to remain competitive. The objectives are primarily increased reliability and productivity. These are to be achieved by greater automation and less overhead such as administration and inventories. The goals are difficult to achieve and require a completely new approach to product design and production. People can manipulate objects on a production line with the equivalent of 200 axes of rotation. The fact that most robot manipulators used in automation have five or at most six such axes leads to fundamental changes in the design of the product itself. Corresponding changes in the design of DCCS implementations for modern manufacturing systems are required.

CIM is characterized by concepts such as: (1) Cell based manufacturing in which families of components are manufactured from beginning to end in a single manufacturing "island" which simplifies product flow, (2) Flexible Manufacturing Systems (FMS) which can be quickly changed to produce different products (frequently used in cells), and (3) Just In Time (JIT) techniques in which costly inventories are reduced by close control and scheduling of resources and processes.

To achieve successful CIM, management must have current access to the status of operations in order to properly control the flow of resources into the facility. The interdependence of all aspects of the production process in JIT systems requires instant reporting and response to problem conditions. As a result of these and other factors, implementation of effective CIM requires computer integration throughout an organization.

Successful CIM operations emphasize the scheduling of tasks and resources. As delays become intolerable and the options for manufacturing a product increase, the scheduling problem becomes sufficiently large that true optimization becomes difficult. Schedulers at best can approximate optimal solutions. Integrated process/management information systems must be able to dynamically respond to changing process conditions in order to achieve high performance. This becomes increasingly difficult when insufficient sensor information exists to specifically identify the source of process anomalies. Consequently, requirements arise on the DCCS for real time software which can emulate the knowledge of an experienced operator who can make decisions in the absence of complete information.

As suggested previously, communications and computer technology is progressing rapidly and can accommodate many of these new requirements. However, the increased complexity of such systems has a corresponding impact on the ease with which a DCCS can be designed, modeled, implemented and verified to be correct. Intelligence, functions, and data must be properly allocated throughout the DCCS. Interconnection schemes must be chosen to eliminate communications bottlenecks. The development of software for such systems is a major management problem. Not only are distributed functions and software more closely related, but deterministic process control logic may no longer apply when decisions must be made with insufficient data. Design of the operator interface is also of increased importance because of greater consequences of errors.

Four areas of activity which will help resolve these problems are now discussed. They are advances in software tools and engineering, computer aided control system design developments, technology to aid operator interactions, and developments in artificial intelligence. Many of these are closely interrelated.

Software Advances

The development of software for large systems has benefited from better languages, operating systems, design methods, and programming environments developed over the last 30+ years. There is enormous interest on the part of the academic, industrial, and government communities in increasing our ability to produce quality software. Despite this interest and a considerable expenditure of funds, software development remains a time consuming, error prone task. While the production of computers and their inherent processing power continues to rise exponentially, the efficiency of software production is only increasing at a rate of about 5% per year. The long term trend is therefore a shift in the cost of large systems from hardware to software.

Modern, General Purpose Languages. Several major languages have been developed to address this problem. These languages, which include such major efforts as Ada, Modula-II, and Praxis[41] were designed specifically to address controls and communications related problems. In addition to providing constructs suited to process control such as synchronization and controlled access to data, these languages emphasize compile time consistency checks on data structures across procedures. This is a recognized common location for errors in multiprogrammer efforts. Such languages are intended

to minimize common errors and to detect errors as early as possible in the development process. Errors in software which has been integrated into complex systems are difficult to identify and isolate. Another goal of these languages is to provide an implementation tool to be used throughout an organization. This leads to decreased maintenance costs by minimizing the requirements for maintaining different staff skills.

Modern languages are accompanied by various types of support environments (e.g., Ada environments) which provide libraries of pretested procedures, communications with other developers, language specific editors and debugging aids, and often specification aids and documentation tools. These languages and environments have all had varying degrees of success. However, they do not impact the software development problem sufficiently to meet the growing needs of complex systems.

Fourth Generation Languages. Several areas of development will impact the software problem. The first simply recognizes that the availability of qualified programming skills at the level required by general purpose, Ada-like languages will not increase fast enough to satisfy demand. The solution is to increase the number of available programmers by reducing the programming skills required.[29] Accordingly, the concept of fourth generation programming languages oriented towards programming by the end user has emerged.

In the DCCS environment, fourth generation languages are being developed which allow less highly trained programmers (such as the process control system operators) to specify functional requirements using process related terms and symbols. Such systems are usually menu driven and/or symbolically operated (e.g., data flow programming).[13] Their output can range from a precise specification of the task to be implemented, to pseudo code which lends itself to straightforward translation into the language of choice. In the limit, self-contained application generators produce tables or instruction sequences which fully implement the designated procedure.

Software Engineering Management. The second area of interest is the development of improved software engineering techniques leading to fewer errors in software and improved system designs.[14] One facet of this effort adapts proven project management techniques from other engineering disciplines to the software environment.[34]

For example, the Work Breakdown Structure (WBS) method of decomposing large systems into manageable "work packages" is being applied with success in many areas (Figure 11).[16] The technique recognizes that the manager must have visibility into all aspects of the software engineering problem including organization, budget, schedule, requirements, design, implementation, configuration management, integration, testing and overall quality assurance. The WBS framework allows successive decomposition of each of these areas to as many levels as are required to reach manageable packages (much like the design of DCCS architectures).

In addition to project management systems, there is intense activity in developing tools to assist the software development effort. Tools which help specify[32] and document functional requirements are among the most important for large projects, but there is also emphasis on tools for writing, simulating, and testing software (e.g., Refine From Reasoning Systems, Inc., and Use.It from High Order Software, Inc.).[15] These tools will benefit both small and large DCCS development effort.

Fault Tolerant and Fault "Free" Software. Another software technique which may receive greater emphasis in the future is fault tolerant design.[25] With such a design approach, software modules or packages are designed to protect themselves from mistakes in other modules using techniques such as extended parameter checking and embedded diagnostics.

Research into formal software validation techniques is also underway. This research is directed at determining methods of formally specifying software functionality and developing corresponding methods of formal design and code analysis to prove correctness. This problem becomes very difficult for all but the simplest modules using current techniques. The time required for proofs increases exponentially with the size of the program or algorithm.[15] However, formal verification techniques offer the promise of being able to prove the correctness of software without exhaustive testing. Such testing itself may be impractical if not impossible, and is generally inconclusive.

Computer Aided Control System Design

CACSD is an emerging field which will also change the manner in which complex, successful distributed computer control systems are developed. Tools have existed for many years to aid in classical control problems such as feedback control, modeling of stochastic dynamic systems, Kalman filter design, etc. Their development is likely to continue, particularly as they are made to be more "user-friendly".

There is new emphasis in CACSD, however, on building upon the aforementioned software development aids by incorporating artificial intelligence techniques (see below), computer aided specification systems, computer aided analysis and simulations, and fourth generation software systems in order to provide a complete computer aided DCCS engineering environment. Commercially available CACSD packages are now appearing which are targeted towards particular classes of processes and particular manufacturer's control system technologies. For example, Data Acquisition Systems offers a package in which the process control requirements are specified using ISA standard symbols in an interactive graphics environment. Control system design and programming is performed automatically.

Tools are also under development which focus on the design of the process itself. The Carnegie Group has, for instance, recently announced the first of a series of process design packages. The first product addresses the process scheduling problem in CIM

operations. Future developments will concentrate on AI-based decision and modeling systems. The trend will be to integrate process design systems with CACSD techniques to provide an integrated solution to process engineering.

Advanced CACSD techniques will ensure the ability of a DCCS to meet performance requirements, will simplify the design process and ensure its manageability, will document DCCS designs and may eventually verify their correctness. High performance graphics will be combined with emerging workstation technology to display the results of simulations using animation. For instance, the movement of robot arms during process operation will be predicted and shown. As CACSD workstations approach 100 Mips performance levels and as the complexity of process control systems increases, icon-driven design and predictive analysis of entire distributed computer control systems will become the norm rather than the exception.

Operator Interaction Related Technology

The operator control station represents the primary human interface with the process itself. The design, performance and functionality of the operator interface is therefore critical to the success of any distributed control system. As DCCS implementations increase in complexity, the challenge is to present sufficient information in a clear manner to properly supervise the process. Design of the operator interface is an important contributor towards correct operation and safety.

Current high resolution color displays are rapidly decreasing in cost and are universally available. High image definition can decrease operator fatigue. High resolution and addressability facilitates the presentation of well organized status and attention displays. Touch panel overlays for CRT's are now commonplace for accepting operator commands. The use of such systems will proliferate to more levels of DCCS' as their embedded intelligence increases and costs continue to decrease.

High levels of process complexity further increase requirements for displaying more information meaningfully. Advances in three dimensional display systems will be the basis of one solution. A recent development using LCD based eyeglasses combined with twin CRT displays for producing stereoscopic images offers a potentially low-cost, high resolution three dimensional display technology. This can be used, for instance, to display 3D physical outlines of the process annotated with appropriate status information. While currently high cost, research in this area combined with low cost fabrication techniques will make the use of such displays feasible where increased clarity of status information is required.

Speech processing technology has matured to the point where both speech generation and speech recognition may be employed in the operator interface. While early speech generation systems were a distraction, the quality of systems has been steadily improved. Speech recognition systems of moderate cost, both speaker independent and dependent, are also available. Speech generation can be useful for attention purposes, but such approaches quickly prove tiresome as similar messages are repeated. A better method for employing speech technology is in an ask/answer mode where operators can request information which is then either verbally or visually communicated. Using speech in the operator interface will allow operations personnel to concentrate more on process characteristics and less on the computer aspects of the control system.

Artificial Intelligence

The increased application of artificial intelligence (AI) techniques to both the design[37] and operation of distributed computer control systems is inevitable. It will be a major contribution to the engineer's ability to meet increasing process requirements. The portion of AI known as expert systems[8] will be covered here, including knowledge-based and logic programming. Other areas such as image processing, pattern recognition, and parallel architectures have been treated previously.

Expert systems frequently combine techniques of logic-based programming languages such as Lisp, PROLOG, OPS-5, and Smalltalk with knowledge acquired from a human expert and translated into data "frames", logic rules, or simply language instructions. The result is intended to emulate an expert's analysis capabilities in a particular area.

Expert systems have found immediate application in the maintenance of complex process components such as robots. Their importance is illustrated by this example from an automobile manufacturing plant in the U.S. A process robot was down for two days after someone accidentally hit the stop button on its controller. No one was able to quickly diagnose the problem as there were hundreds of possibilities. After two days, an experienced person arrived who was able to quickly discover the problem, based on his familiarity with the product. This is a natural area of application for expert systems. Their use will certainly be extended in the future to other areas of DCCS maintenance. Incorporation of such systems into the DCCS itself will be desirable.

Expert systems can analyze incompletely characterized problems using logic programming techniques. If a conclusion is to be reached, the result is characterized by its likelihood and the reasoning process used can be examined by the user to aid in verifying the result. Accordingly, expert systems can be used to analyze improperly functioning complex processes and give recommendations as to action. Alternatively, real time expert systems are now under development which will allow their use in on-line, fully automatic DCCS functions. Expert systems will be an essential contribution to the analysis and control of large processes where, by one estimate, the addition of only 20 sensors can double the complexity of the control requirement.[18] Expert systems can also be applied to other areas of complexity, such as scheduling for CIM systems and optimization of DCCS designs.

An important issue in expert system programming is the methodology behind knowledge engineering. The knowledge engineer has particular skills in translating the implicit and explicit knowledge of an expert into the appropriate computer representation. Knowledge engineering is an emerging discipline which will need to be represented in future DCCS development teams.

Issues Which Will Affect Future DCCS Systems

There are emerging non-technical issues will influence the evolution of DCCS systems and their component technologies. The options and criteria facing individual system designers will be affected. In this last section we will consider the impact of these issues.

Education

The availability of people trained in DCCS design has until recently depended almost exclusively on people with fundamental electronics and computer skills who have received "on-the-job" training. This is beginning to change. The recent world-wide emphasis on manufacturing automation has encouraged institutions to establish laboratory environments for graduate studies in CIM (e.g., Imperial College of London, University of Swansea, U.K; University of Witwatersrand, S.A. and Draper Laboratory (MIT), U.S.A.). More recently, undergraduate CIM programs in smaller institutions such as the Oakland Community College near San Francisco have been established. As these programs mature, more trained personnel will become available. Moreover, the academic influence will help to construct long range strategies for CIM and corresponding DCCS development, which is currently being driven (productively) by nearer term industrial objectives.[6]

Economic

Manufacturers are in an automate or perish situation with regards to world-wide competitiveness. Strong long term economic considerations are driving the current push toward automation. This is being led by the largest companies with the most capital to invest. Paradoxically, as long as economic conditions continue strong, this investment is likely to continue. However, emphasis on developing systems which can be economically utilized by smaller companies must be increased if the rapid product innovations characteristic of these companies are to continue.

Sociological

Sociological issues of both personal and community origin will affect future DCCS design parameters. Recent instances of large system failures (e.g., Challenger, Three Mile Island, Chernobyl) have increased public concern over the safety and quality of increasingly complex systems. Public perception is a powerful force. Poor shift against high technology efforts could result if stringent engineering quality standards are not emphasized. This would seriously weaken support for advanced large scale research efforts and could slow down world-wide efforts to improve standards of living.

With the current visibility of large, high technology systems, individual DCCS engineers have a serious and somewhat intimidating responsibility to ensure the highest quality in their efforts. In addition to more rigorous design procedures, developers may need to increase the use of archival records to clearly identify the course of failures in order to ensure their future prevention. Other impacts on DCCS design are likely.

Engineering Ethics

There is an increased awareness of the importance of personal ethics in all aspects of society. As organizations grow and the management structure becomes more remote, individual responsibility becomes increasingly essential in the quality of the product. The ability of organizational structures and rules to ensure ethical practice, responsibility, quality, and safety is at best limited. Such characteristics must come first from the individual.

Several organizations are now starting formal training courses to help employees develop a sense of ethical awareness. These courses are likely to increase in number. The impact of such awareness on issues related to safety and quality may require new paradigms for interactions between engineers and organizations.

Political

We conclude this section with a note on political influences on future DCCS systems. Economically and socially motivated trade barriers and sanctions are factors in the world today. Such artificial barriers may or may not be appropriate. However, wherever they exist, two things are likely to occur. First, if a country has access to other technologies capable of performing the necessary functions, and subsequently adopts these in lieu of protected or embargoed technologies, there is a gradual economic shift favoring the accessible technology. Such a technology may prevail even if it is inferior. This may limit development of the superior technology and eventually limit future design options.

Second, in the case of severe technology embargoes, countries are encouraged to develop their own technologies and standards. This can lead to technological independence for much longer periods of time than the embargo lasts. The redevelopment of technology markets in such countries becomes a long term task which many companies will not even attempt. In the extreme case, such countries may develop competitive technologies that substantially impact other world markets. The impact here is

competing technology choices available to developers. Within technologically isolated countries, conversely, the technological options become much more constrained.

Summary

This paper has reviewed the major aspects of distributed computer control system design and implementation. The DCCS engineering environment is seen as requiring a broad range of technologies from the computer and electronics engineering fields. The task of creating optimum designs for DCCS' is an unsolved problem for complex systems. Performance prediction and assurance of correct operation is being accomplished through a combination of analysis, modeling, simulation, and prototyping.

The DCCS field is evolving to meet increased functional needs and performance requirements related to large scientific processes and complex commercial processes. A major influence is the global thrust towards computer integrated manufacturing.

Several emerging technologies are likely to influence future DCCS designs. Advances in communications and VLSI technology will lead to DCCS architectures characterized by highly distributed intelligence down to the process sensor level. Hierarchically connected systems will be replaced by uni-level designs in which a hierarchy of process computers communicates over one or a limited number of bus-oriented communications systems. Image processing functions are expected to become pervasive, particularly in CIM applications. Emerging low cost PC and LAN technology will fill the continued need for economical implementations of moderate performance distributed control systems.

Emerging artificial intelligence techniques and computer aided control system design packages will become essential elements of future DCCS designs and implementations. As systems increase in size, public visibility, and cost, there will be increasing emphasis on fault tolerant and provably correct software.

The overall prognosis for the DCCS field is for rapid growth in demand and performance. This will be met by incorporating the most advanced technologies available in computer hardware and software, large scale integrated circuit production, and industrial automation. Engineering designers will continue to be challenged by increased requirements, emerging technologies, and socioeconomic issues in the future.

Acknowledgments

The authors wish to acknowledge the many informal and formal discussions with participants at the 1985 IFAC workshop on Distributed Computer Control Systems which led to many of the ideas presented here. In addition, informal conversations with Frederick W. Holloway (LLNL) during the preparation of this manuscript provided important clarifications and contributions which are gratefully appreciated.

References

1. Roger Allan, "Factory Communication: MAP Promises to Pull the Pieces Together," Electronic Design, May 15, 1986, pp. 103-112.

2. Karl Johan Astrom, "Process Control - Past, Present and Future," IEEE Control Systems Magazine, August 1985, pp. 3-9.

3. Brent Auernheimer and Richard A. Kemmerer, "RT-ASLAN: A Specification Language for Real-Time Systems," IEEE Transactions on Software Engineering. September 1986, Volume SE-12, pp. 879-889.

4. Michael Babb, "Using Fiber Optics to Communicate with Factory Floor Devices," Control Engineering, August 1986, pp. 39-41.

5. K. L. Boyer, A. J. Vayda, and A. C. Kak, "Robotic Manipulation Experiments Using Structural Stereopsis for 3D Vision," IEEE Expert, Fall 1986, pp. 73-94.

6. Edgar H. Bristol, "An Industrial Point of View on Control Teaching and Theory," IEEE Control Systems, February 1986, pp. 24-27.

7. Harold W. Carter, "Computer-Aided Design of Integrated Circuits," Computer Design Automation, April 1986, pp. 19-37.

8. B. Chandrasekaran, "Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design," IEEE Expert, Fall 1986, pp. 23-32.

9. Alan E. Charlesworth and John L. Gustafson, "Introducing Replicated VLSI to Supercomputing: the FPS-164/MAX Scientific Computer," Computer, March 1986, pp. 10-29.

10. Department of Defense (DOD) Reference Manual for the Ada Programming Language, Proposed Standard Document, United States Department of Defense, 1980.

11. D. G. Dimmler, R. Fine, and C. Levinthal, "Fastrun - A High Performance Computing Device for Molecular Mechanics Using a Pipeline Architecture, October 1985.

12. Stanley B. Gershwin, Richard R. Hildebrant, Rajan Suri, and Sanjoy K. Mitter, "A Control Perspective on Recent Trends in Manufacturing Systems," IEEE Control Systems, April 1986, pp. 3-15.

13. Ephraim P. Glinert and Steven L. Tanimoto, "PICT: An Interactive Graphical Programming Environment," Computer, November 1984, pp. 7-25.

14. Hassan Gomaa, Software Development of Real-Time Systems," Communications of the ACM, July 1986, Volume 29, Number 7, pp. 657-668.

15. Margaret H. Hamilton, "Zero-Defect Software: The Elusive Goal," Spectrum, March 1986, pp. 48-53.

16. Norman R. Howes, "Managing Software Development Projects for Maximum Productivity," Software Engineering, January 1984, Volume SE-10, pp. 27-35.

17. C. Imamichi and A. Inamoto, "Uni-Level Homogeneous Distributed Computer Control System and Optimal System Design," Distributed Computer Control Systems 1985, DCCS-86 Proceedings, pp. 133-144.

18. Rod Khanna, "Sound the Alarm: Expert Systems Will Respond," Computer Design, pp. 90-91.

19. Mansur R. Kabuka and E. S. McVey, "Input-Output Characteristics for Imaged Transducers," IEEE Journal of Robotics and Automation, June 1986, Volume RA-2, pp. 106-115.

20. Gadi Kaplan, "Industrial Electronics," IEEE Spectrum, January 1986, pp. 61-65.

21. Ernest W. Kent, "Eyes for Automatons," IEEE Spectrum, March 1986, pp. 37-45.

22. Peg Killmon, "Design Tools Move Towards 21st Century," Computer Design, July 1, 1986, pp. 51-72.

23. Peg Killmon, "Optical Disk Drives Promise to Revolutionize Computer Use," Computer Design, August 1, 1986, pp. 56-64.

24. S. D. Kraft and Edward T. Wall, "Experimental Microprocessor-Based Adaptive Control System," IEEE Control Systems, February 1986, pp. 35-39.

25. Jon G. Kuhl and Sudhakar M. Reddy, "Fault-Tolerance Considerations in Large, Multiple-Processor Systems," IEEE Computer, March 1986, pp. 56-67.

26. J. M. Lane, "Digital Computer Control of a Catalytic Reforming Unit," ISA Trans., Vol. 1, 1962, pp. 291-296,.

27. M. Mariton, "Local Area Networks and Future Process Control Systems," IEEE Control Systems, April 1986, p. 27.

28. Ken Marrin, "Packet-Switched Communications Extend Multiprocessor's Range," Computer Design, September 1, 1986, pp. 20-23.

29. James P. McGill, "The Software Engineering Shortage: A Third Choice," Software Engineering, January 1984, Volume SE-10, pp. 42-48.

30. S. P. Meyer, I. M. Macleod, and M. G. Rodd, "The Design of 'Shoptalk', A Distributed Multiprocessor Computer Control System," DCCS-86 Proceedings (published in this Proceedings).

31. N. Mokhoff, "Design and Engineering," Computer Design, July 1, 1986, pp. 64-73.

32. Peter Gabriel Neumann, "On Hierarchical Design of Computer Systems for Critical Applications," IEEE Transactions on Software Engineering, September 1986, Volume SE-12, pp. 905-920.

33. David Nitzan, "Development of Intelligent Robots: Achievements and Issues," IEEE Journal of Robotics and Automation, March 1985, Volume RA-1, pp. 3-13.

34. Arnold M. Ruskin and W. Eugene Estes, "What Every Engineer Should Know About Project Management", 1982, published by Mercel Dekker, Inc.

35. Sydney F. Shapiro, "Jumping on the MAP Bandwagon: How Soon and How Far," Computer Design, August 15, 1986, pp. 9-21.

36. A. Silbey, V. Milutinovic, and V. Mendoza-Grado, "A Survey of Advanced Microprocessor and HLL Computer Architectures," Computer, August 1986, pp. 72-85.

37. Herbert A. Simon, "Whether Software Engineering Needs to Be Artificially Intelligent," IEEE Transactions on Software Engineering, July 1986, Volume SE-12, pp. 726-732.

38. T. M. Stout, "Computer Control of Butane Isomerization," ISA Journal, Vol. 6, 1959, pp. 98-103.

39. Gregory J. Suski, "Distributed Computer Control Systems 1985," Proceedings of the Sixth IFAC Workshop, Monterey, California, USA, May 20-22, 1985.

40. Gregory J. Suski, Editor, "Round Table Discussion," Distributed Computer Control Systems 1985, Proceedings of the Sixth IFAC Workshop, Monterey, California, USA, May 20-22, 1985, pp. 219-226.

41. Gregory J. Suski, F. W. Holloway, and J. M. Duffy, "Experience With a High Order Programming Language on the Development of the Nova Distributed Control System," Distributed Computer Control Systems 1983, pp. 73-84.

42. John Voelcker, "Helping Computers Communicate," IEEE Spectrum, March 1986, pp. 61-70.

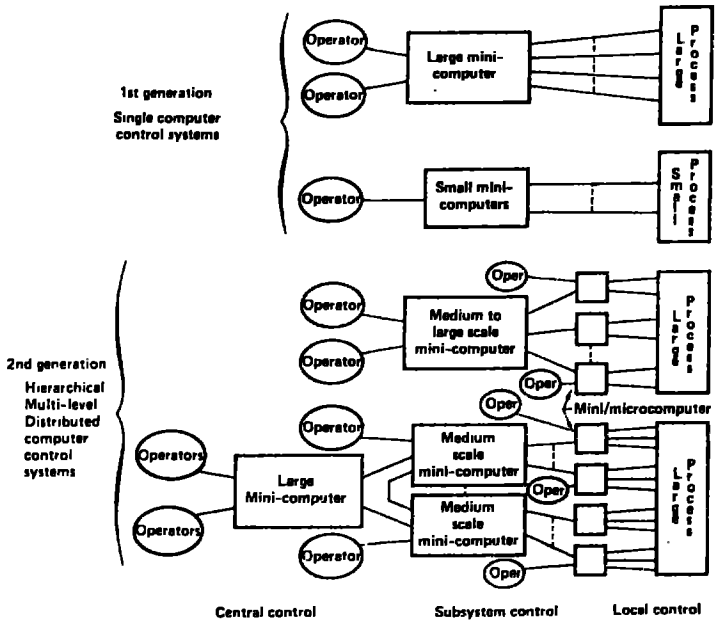


Figure 1. Computer-based control systems, first and second generations.

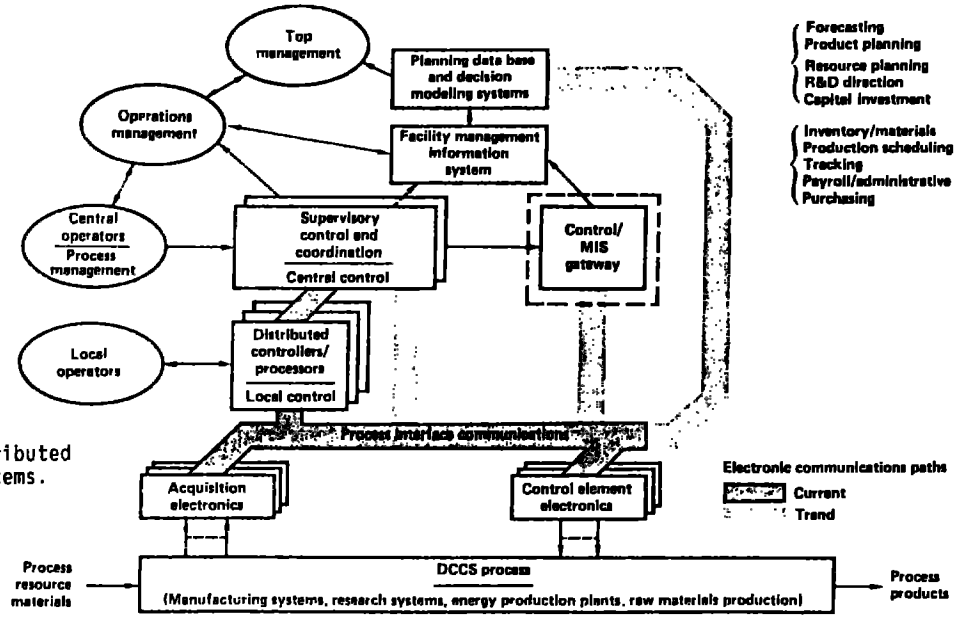


Figure 2. A model of distributed computer-based control systems.

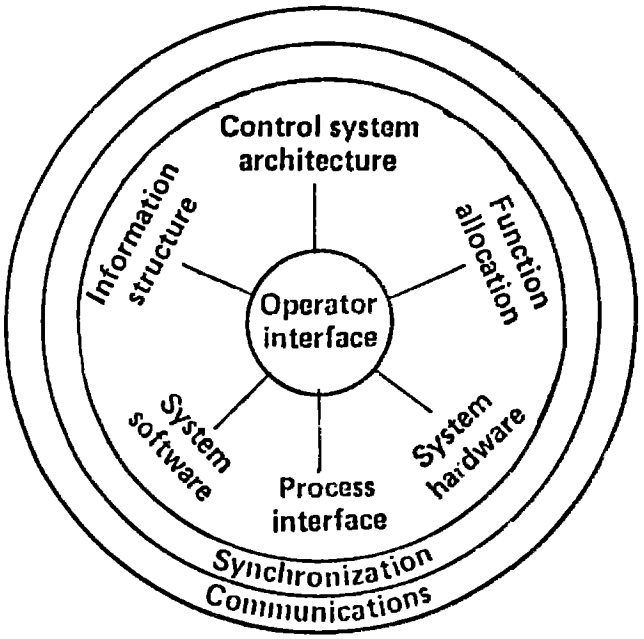


Figure 3. Design considerations for distributed computer control systems.

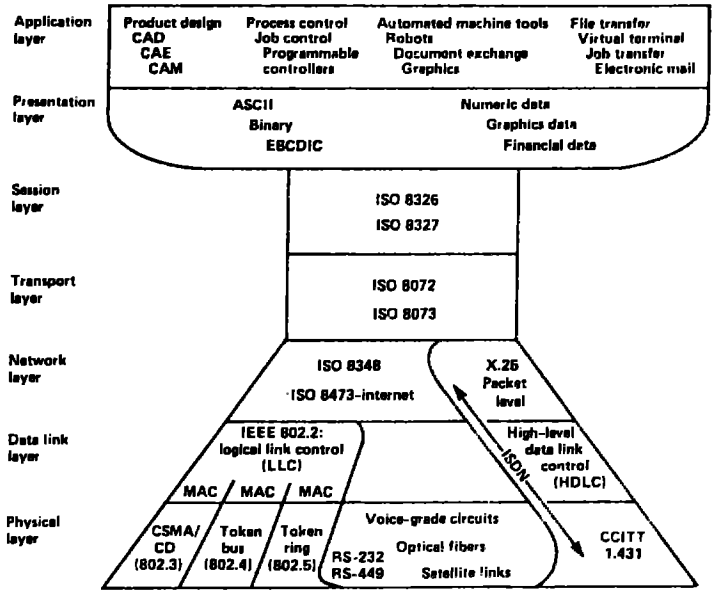


Figure 4. Communications standards based on the ISO Open Systems Interconnect (OSI) Model are emerging (adapted from Voelcker, IEEE Spectrum, March 1986, pp. 64).

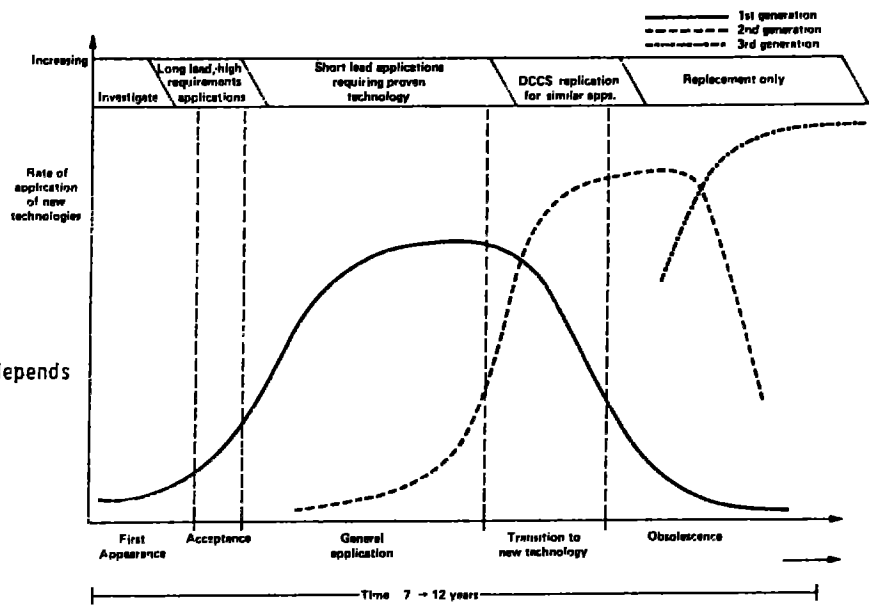


Figure 5. Selection of technology depends on its maturity and requirements of application.

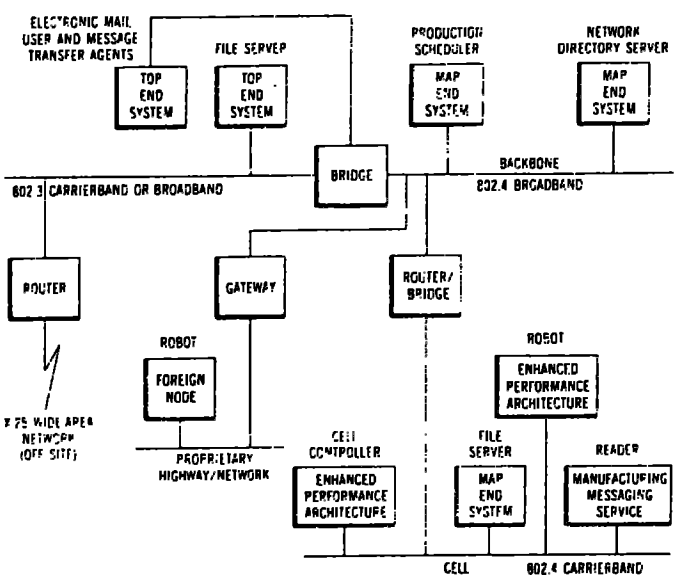


Figure 6. Version 3 MAP and TOP protocols are likely to support carrierband and broadband protocols.

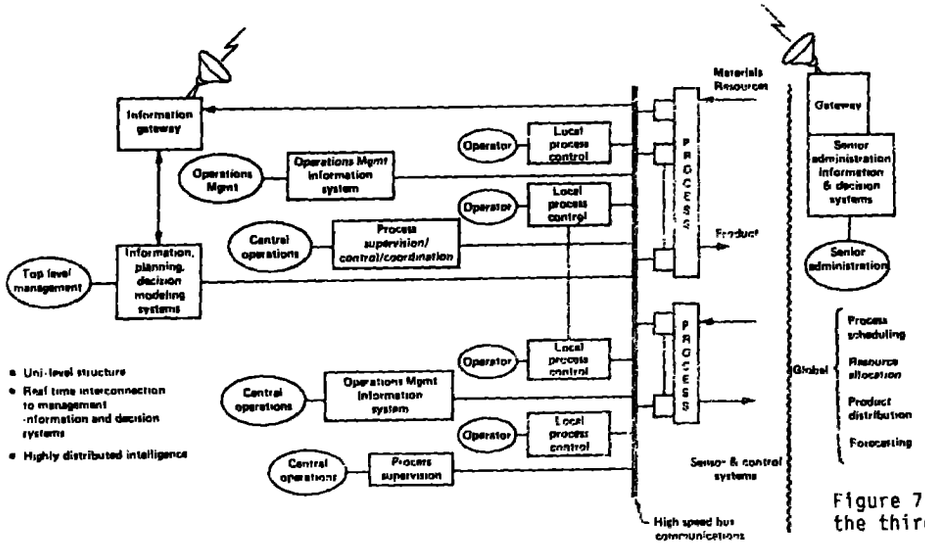


Figure 7. Computer-based control systems, the third generation.

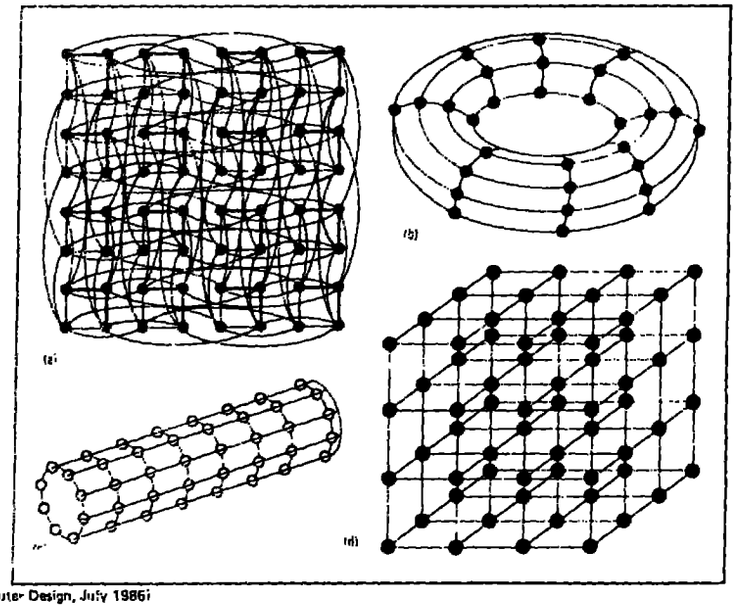


Figure 8. Parallel computer architectures: hypercube, torus, cylinder, and mesh.

(From Computer Design, July 1986)

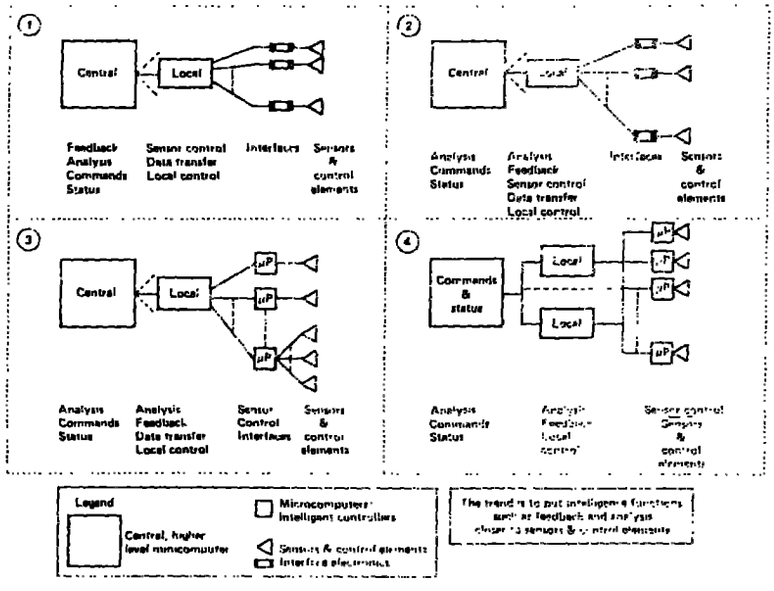


Figure 9. Intelligence becomes more distributed in control systems as computer technology evolves.

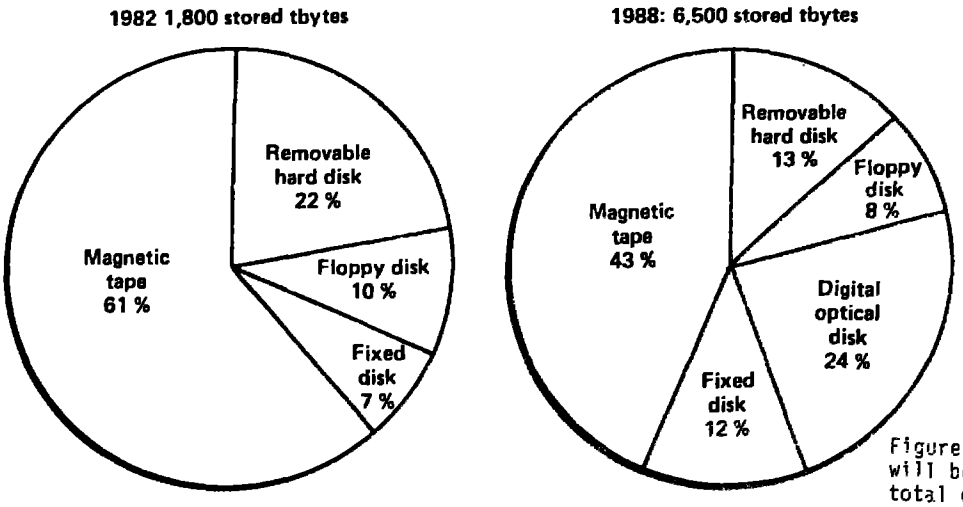
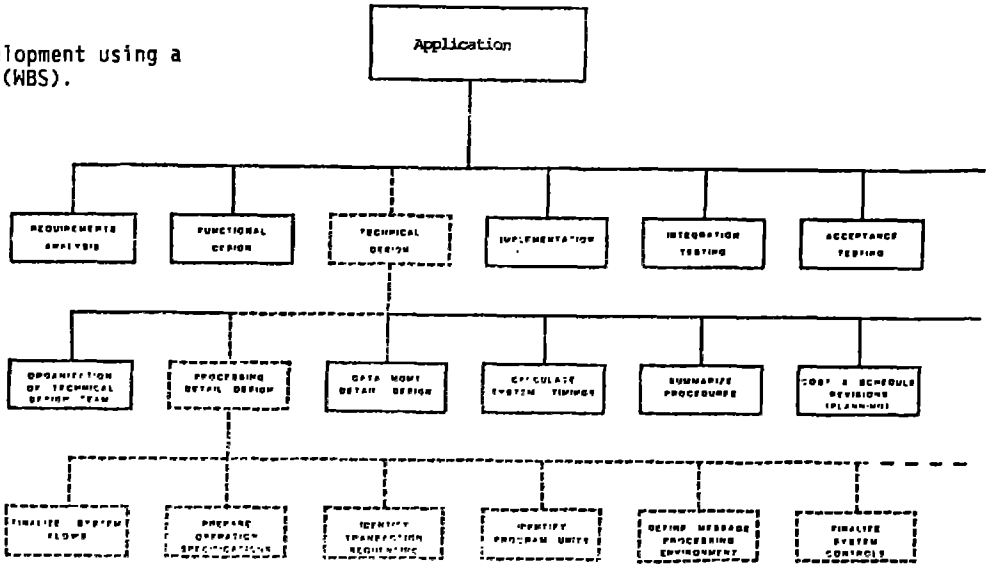


Figure 10. Use of optical disk technology will become a major contributor to our total data storage capacity.

(Source: Computer Design, August 1, 1986)

Figure 11. Software development using a work breakdown structure (WBS).



(from Bowes, IEEE Transactions on Software Engineering, January 1984)