C. Kinberg
B. W. Landeck

Integrated Manufacturing Systems: Architectural Considerations

Abstract: This paper is divided into two main parts. The first is concerned with definitions of concepts and terms of importance to the architect of integrated manufacturing systems. It describes the operational environment, how the manufacturing systems are going to be used, and a variety of system requirements. Priorities are established and a set of guiding principles for system design are proposed. In the second part, three basic system structures are analyzed and evaluated in terms of the previously stated priorities and guiding principles. The relation of integrated manufacturing systems to other complex systems is shown to illustrate how generally applicable many of the findings derived from knowledge of the manufacturing environment are to the design of systems for other application fields.

The paper concludes by restating the key requirements and formulates what appears to be the greatest challenge for the system architect. That is, the creation of an architectural framework within which the system designers can structure manufacturing systems capable of handling besides the planned tasks a variety of unforeseen applications. Readers familiar with the plant environment and the terms and concepts associated with it may skip the first sections and start with System Requirements. We would, however, advise even the experienced reader to review the definition of real-time systems, which we believe is new.

Introduction

Several papers[1–5] in this issue of the Journal are concerned with reporting on achievements in the application of computer systems to manufacturing problems in IBM. This paper aims at generalizing experiences gained within IBM over a period of some five years and at expressing the outcome of the analysis in terms that are relevant to systems architecture. Significant information from customer installations has also been taken into account. The purpose of this paper is to give a comprehensive picture of the factory operations environment, the computer-based automation efforts, and important requirements that have to be met by systems and users. It is not intended as a system design guide.

Manufacturing control systems are important subsets of the total plant automation systems. Since they interact intimately with other system components a review of architectural requirements should be concerned with all computing activity in the plants. To enhance the understanding of subsequent sections, we offer the following definitions:

C. Kinberg and B. W. Landeck are located at the IBM Systems Development Division Laboratories in San Jose, California and Poughkeepsie, New York, respectively.

System architecture: The development of a system architecture is the process of melding a variety of individual disciplines related to hardware design, programming, application design, communication, man-machine interaction, etc., into a cohesive description of a system or a product line. Thus, a system architecture, as such, is a detailed description of interfaces between system components and of component capabilities reflecting the results of judicious trade-offs among disciplines.

To guide the architecture design, a set of well-defined functional objectives must be established. By proper formulation of these objectives, the requirements of particular application areas such as plant-floor control can be taken into account. Hence, the procedural steps in the evolution of a system architecture must start with the interpretation and organization of application requirements to be expressed in suitable terms, e.g., architectural objectives.

The architecture development process culminates in the creation of a set of system architecture specifications that define interfaces, implementation mode (hardware or software), data formats, instruction sets, etc. These specifications serve to guide the design of products and

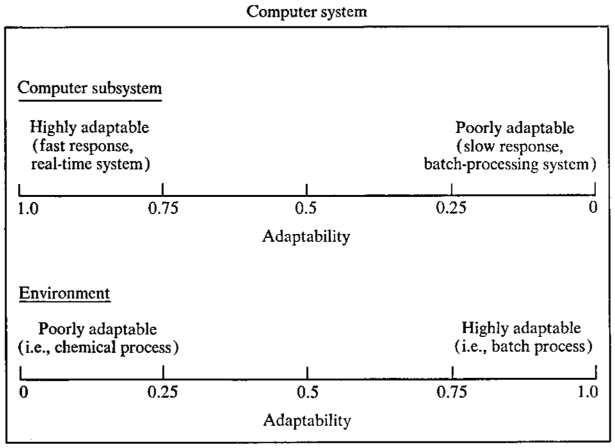


Figure 1 Adaptability scales for efficient matching of computer subsystems to processing environment.

systems. Once generally accepted, the system architecture specifications provide the yardstick against which actual system implementations can be measured. The review of requirements that follows the description of the manufacturing systems environment shows that a key criterion for good architecture design is the amount of flexibility offered to the implementers of systems for different application areas.

Integrated Manufacturing Control Systems play an essential role in the process of automating the factories—a task that will be a preoccupation of the industry for an unforeseeable time. It is important to distinguish between "automation" and "mechanization." Initially, the two words were used interchangeably to describe (among other things) the process of producing or reproducing by machine an effect normally produced by man. The technique has been successfully applied to a variety of individual mass production processes and has often been referred to as "Detroit automation." We shall call this technique "mechanization" and extend the concept of "automation" to encompass the combination of at least partial mechanization with overall operations control of manufacturing activities. Mechanization has some importance since it affects the way in which the computer systems interact with the production processes.

Plant automation: In this paper plant automation is understood to be the application of fully or partially automatic equipment for the purpose of aiding, complementing, amplifying, or replacing human activity in the manufacturing process and its associated information processes. It is and is likely to remain a gradual, technology-based process aimed at increasing productivity and at reconciling the ever-increasing wage levels and diminishing skill resources with the demands for more reliable, more complex, and more varied products.

It has been recognized that the automation process must be dealt with as a systems problem and that computer systems of many kinds will constitute the basic building blocks for linking a practically infinite number of subtasks into coordinated activities. The automated plant need not be a fully mechanized plant, but it is going to be a programmed facility.

In the course of the efforts of the systems architect to develop efficient, streamlined solutions, he must, however, never lose sight of the fact that, whatever application environment he selects to formulate the needs, his architectural concepts must be stated in terms as general as possible to ensure broad applicability of the system structures.

In a later section, we shall review the relation of manufacturing systems concepts to solutions for other application fields such as data-base systems (including management information systems) of which manufacturing systems ultimately will form subsets, educational systems, hospital information systems. The identification of common elements among the different sets of functional requirements is an important task.

Real-time systems: We shall define a "computer system" as the coupling of a computer (e.g., an IBM 1130 System) with an environment (such as a small scientific shop). There are many combinations of computer subsystems and environments, but only a few are practical. We can arrange or group the set of possible environments in a logical order and we choose to do this by ordering them on an "adaptability scale" (see Fig. 1). In a similar way we may arrange the computer subsystems according to their abilities to adapt to the environments. An example of a highly adaptable environment would be a commercial batch shop. The operator can predict how soon he must start a payroll processing job in order to complete it on time. Thus, he, as part of the environment, is adapting to the computer subsystem. When coupling a computer to a chemical process, it is the responsibility of the computer subsystem to adapt to the environment since the chemical process has but a very limited ability to adapt.

Assuming for the moment that the adaptability levels of a computer subsystem and an environment could be measured or assessed, this would allow us to plot specific subsystems and environments on a zero-to-one scale and to determine in a simple way which systems are "functional," i.e., capable of performing the desired tasks. Thus, in order for a computer subsystem to be functional, the sum of the adaptability ratings would have to be equal to or greater than one. Also, it is apparent that the greater the sum is than one, the more inefficient or the more unused capability the computer subsystem has. An example would be the use of an IBM 1800 System in a pure batch environment. The system will do the job, but much of its capability would be unused. An IBM 1130 System (which is essentally an 1800 with much less adaptive capability) will perform the same job much more efficiently, i.e., at lower cost. We do not know at present of any accurate method of assessing adaptability quantitatively.

Real-time computer subsystems are those computer subsystems which offer a high degree of adaptability. The environments to which they are coupled range from only slightly adaptable to completely inflexible. The most inflexible environments are those involving the operation of mechanical and chemical processes. When man is added to the environment its adaptability usually increases.

In control-oriented environments, we can be quite specific:

1) A real-time subsystem must have an inherent capability to adapt to changes in its environment and in its own structure. Here, "adapt" may imply switch of task and/or switch in functional architecture.
2) For real-time subsystems to be "functional" in a given environment the feed-back delay must be small compared to the smallest time constant of the environmental variables. Feedback is here meant to be the result of some processing step(s) needed for proceeding with work.

One of the prime parameters of computer subsystem adaptability is called "response time." This parameter is a measure of the time required to respond to a "happening" in the environment. Normally, the shorter the response time of the computer subsystem, the better it is able to adapt to changes in the environment.

We define a system as operating in real time when the computer subsystem is adapting to changes in the environment as they occur in such a way that the environment does not need to be conscious of its connection to the computer subsystem.

Hierarchy concepts

Since one of the purposes of a system architecture is to provide a framework within which a customer installation can grow harmoniously toward the ultimate goal, a well-functioning total-system solution, it is very important to have full appreciation of the hierarchy concepts according to which modern organizations are evolving.

Organization begins when any group of individuals joins forces for a specific purpose. The basic principle governing the organization process is coordination, which we define as the orderly arrangement of group effort. The foundation of organization is authority, a right which is always associated with a set of responsibilities. The coordination principle has a subordinate called "the scalar principle," which controls the forming of organizational hierarchies. The scalar process involves the grading of duties according to levels of authority and responsibility. It is also the process by which the coordinating authority becomes effective throughout the organizational structure.

We now analyze the pattern of typical activities following the traditional hierarchy within a factory. The top level is the business planning and office automation level where conventional data processing activities are carried out, such as accounting, inventory control, invoicing, purchase order processing, etc.; strategic planning, forecasting, and budget work are other activities on this level.

Immediately beneath this comes a level of activity associated with operations planning and control. This activity is concerned with production planning, making the best use of available and expected resources and facilities within the framework fixed by market forecasting and budget. The information on capability is then made available to the sales department for quotation of prices and delivery dates. As definite customer orders are received, a production schedule can be established. This time table will probably be revised from time to time as forecasts change and better estimates of the capabilities become available. The updated production schedule is then used to provide detailed orders for the production facilities and the purchasing department at appropriate times. Detailed objectives for the individual manufacturing processes can also be derived. Thus, certain processes should be operated to incur minimum costs, others to maximize yield, some to minimize use of intermediate products in short supply, etc., according to how the operation of each process affects the factory and its economics as a whole.

The third level, concerned with operation of the individual manufacturing processes, is thus fed from above with all the necessary information about targets and time schedules. Information on actual performance and accomplishments on this level is reported back to the next higher level and the results are used by the scheduling function in revising the time tables for the next production period. Thus, failures to make the required schedule must be rectified, overproduction of certain materials must be recorded and future orders amended. Possible surplus capacity should be notified to the sales department and other pertinent information accumulated for the generation of better estimates of the performance of the various production facilities.

Since the nature of the jobs in the factory changes from level to level in the hierarchy, it is not surprising that the requirements placed on manufacturing systems show a corresponding differentiation as we move vertically through the organization. The stronger this differentiation is, the more likely we shall find system solutions structured as hierarchies of computer systems. It is not surprising that such hierarchies of computer systems have a tendency to image organizational structures.

Operational environment

The answer to the question "what requirements should be placed on integrated manufacturing control systems?" can be found after we have determined how and for what tasks the systems are going to be used.

As a general statement, we propose that the main function of the future systems will be to aid and support a major portion of the individuals in an enterprise in their daily work. Today, only a very small part of the personnel of a company is relying on direct support from the computing center. We expect some of the greatest changes in computer usage to be related to who makes use of the systems and how they are being applied. A first consequence of the increasing dependence on computer support will be that the central computing shops will have to be complemented with terminals physically located in practically all areas of the factory. A second consequence is that in the future a large portion of the users will have only rudimentary knowledge of programming and computer application technology.

The number and variety of tasks which the fully integrated manufacturing systems must be capable of handling are great indeed. They cover practically the total application spectrum to which computer systems are being applied today, including batch processing, data-base related jobs, various real-time sensor-based and operator-oriented functions. To provide solutions that can satisfy such a range of requirements the architecture must conveniently accommodate at least two major categories of subsystems, capable of performing information system functions and equipment control system functions, respectively.

There are many methods of subdividing the jobs we expect the computer systems to perform and assist the operators in executing. An interesting way of analyzing the jobs would be to distinguish between work performed in centralized computer facilities operating in a closed-shop fashion and data-base operation oriented tasks executed in conversational mode. The former type of operation involves generation of large numbers of reports regularly and mostly according to fixed formats. The latter category of tasks, which will be increasingly important in the future, involves direct interaction between many users and the system and introduces a new dimension to problem solving and management which will not be discussed much further in this paper. In due time, this second mode of operation will have strong impact on system architectures. Since a thorough analysis of the implications would have to be very extensive, we have elected to review the categorization of jobs from another angle. From a requirements point of view, it makes sense to review information system functions and equipment control functions separately. (Since all computer systems are information systems, we are not satisfied with the terminology. The choice of categories is, however, useful for the analysis so we ask the reader to please bear with our semantics.)

Note that although in the future an increasing and eventually dominating number of tasks will be executed in conversational man-machine interactive mode, many jobs will be more efficiently handled in batch mode.

• *Information system functions*

Continuing the categorization, we group, as follows, the information system functions according to which particular segment of the factory they support:

Management information system functions: This set of functions supports managers on all levels in the execution of their assigned responsibilities. The basic purpose of the management information system function-set is to improve the quality and timeliness of decision making across the factory. System requirement will vary greatly by management level, nature of operations, and by individual preferences.

The systems must be highly adaptable (in the sense of our definition of adaptability) to provide services in real-time since many jobs will be executed in an interactive mode. Thus, the managers should have direct and immediate access to high-level summary information on the status of the plant or segments thereof. Furthermore, they should be allowed to request and obtain more detail when particular problems call for analysis and decision making. In due time the systems will attain a level of sophistication that permits accurate analysis of cause and effect relationships. Ultimately, the managers will be aided in their daily chores by systems capable of simulating the effect of different decision alternatives—a capability that promises to improve the quality of management on many levels.

Planners of future manufacturing systems must, however, make sure that the control efficiency and effectiveness of the operational systems (i.e., those that fall within the areas of production support and equipment control) be raised to a level high enough to ensure that the management information systems can be used mainly as sophisticated planning tools and less as management instruments for trouble shooting.

In summary, we see the management information systems evolving into data-base, real-time, and communication-oriented systems with far-reaching demands on efficient man-machine interaction.

General support functions: These functions will handle a major portion of future batch-processing jobs in the factories. The activities are not directly related to the product flow in the plant, but are necessary for the operation of the factory. Functions performed are employee and supplier payments, cost reporting, cost estimating, pricing, tax reporting and various other administrative tasks. The transactions, such as updating and report generation, are mostly scheduled in advance and the users are in general satisfied if system turn-around time is matched to the plant operation cycle time.

Production support functions: These functions form the information-processing part of the operational systems referred to above (the other part is covered by the equipment control functions). The production support functions are concerned with the gathering, processing and dissemination of large volumes of information related to a product throughout its lifetime, i.e., from the moment it is conceived until the last item has been delivered to a customer. (Actually product data in some form may continue to come in until the last service call has been made.)

The major purpose of this set of functions is to directly control the critical path of the products from the designer's drawing boards through final test and shipment of the finished products to the customers. The large variety of tasks includes the preparation and dissemination of information from the designer to the model shop and manufacturing floor, the processing of material and tool orders, the generation of schedules and time-tables, dynamic allocation of men and machines and the maintaining of an up-to-date inventory of parts.

Typical for this application environment are the ever-changing tasks, the high frequency of large or small data alternations which may affect many functions in the chain of jobs, and rapidly swelling data-banks.

The importance of having a highly efficient product support system cannot be emphasized enough since it will contribute to solving the most difficult problems in the plant, the processing of the products through the facility. Thus, an adequate production support system can effect significant improvements in the following ways:

- Shorter product processing time (product flow time),
- Reduced shop schedule preparation time,
- Reduced inventories: material and parts,
- Tighter production control,
- Better resource utilization.

The success of a production support system depends on the availability at all times of complete and up-to-date information necessary for controlling plant operations in real time. Also important is high flexibility to ensure fast and easy adaptation of the system to changes in functions required. The implications are a need for advanced data files, terminals, an efficient communication system and sophisticated programming support.

The functions within the three areas complement each other. A key requirement for satisfactory overall performance is that information available to the systems be both accurate and up-to-date and available to all subsystems without delay.

The organization and building of the data-base in the factory environment will be very critical. To satisfy the demands for fast access, data security, integrity, and consistency calls for the development of a storage architecture which can accommodate a hierarchy of databanks. The data-banks will provide one major coupling element between the various subsystems forming an integrated manufacturing system.

• *Equipment control functions*

With exception of some special equipment for efficient man-machine interaction (terminals) and associated programming, practically all functions discussed above can be handled by future general-purpose products which will all have some real-time capability. The functions to be discussed now require additional capabilities which should not necessarily be "imposed" on the standard products. The special features required contribute to making the equipment control systems more "adaptable."

The application areas for which real-time oriented "adaptable" subsystems are required pertain to three main task categories to be referred to as *Make*, *Move* and *Test*. These jobs have the flavor of discrete processing, but we must point out that these application categories may involve continuous processes.* Consequently, most plant floor requirements discussed in this paper reflect both continuous and discrete process needs.

The equipment control systems are actually process control and data acquisition systems. In the future they will be provided with features to facilitate their integration into the complex hierarchy of computer systems that form integrated manufacturing systems. The equipment control systems have two basic characteristics that enable them to furnish the link between the complex of computer systems and the real world of instruments and machines:

- Their real-time capabilities enable them to respond adequately to changes in the physical or operating environments.
- They provide interfaces to many types of instruments

* There are some important differences in the environments of continuous and discrete processing: In the oil industry the variety of products generated by each (continuous) process is limited and a major element of automation is the controlling and optimizing of individual processes. In the metalworking, electronics, and aerospace manufacturing industries the variety of products is large and rate of product obsolescence is very high. Thus, total mechanization will probably never happen and the automation process will be based on providing a sophisticated overall-control capability.

and machines and provide the means for reading and disseminating signals of great variety.

Make: There is an almost unlimited variety of machines and processes for making, treating, refining, and finishing parts and materials.

Among machines for discrete fabrication, the following types are amenable to on-line computer control: metal-forming machines, cutting machines, furnaces, coilwinding machines, polishing machines and automatic wiring machines. This list is by no means exhaustive.

Numerical control of machine tools opened the way for direct computer control of a variety of production machines. The extension of the capabilities to include adaptive control promises better yield, higher production rates, and improved quality, but will increase the demands on the computers with respect to real-time performance.

Move: Material handling is the major activity in the move task. In the most basic form of automating this task a computer just keeps track of all components and assemblies during production (the production support function). The task includes the transfer of in-process parts between workstations and the transportation of material, parts and assemblies into and out of the warehouses. Being in essence unproductive, the amount of material handling done should be minimized. In mass production oriented plants mechanization is achieved by the installation of transfer lines. In job-shops, we notice a trend toward circumventing the problem by using multipurpose machines (machine centers).

The mechanization of material handling has been very slow, partly because of the difficulties associated with developing solutions capable of handling different types of parts and partly because the transfer process may include part orientation. Although computer control of manipulators and transfer lines will gain acceptance, the major benefits of the computer systems will be derived from better scheduling of the transfers between work stations.

Warehouse automation has considerable potential because the economic benefits are substantial. The main difficulties are the need for extensive custom design and the high level of system analysis effort required.

The assembly function is a combination of "make" and "move." Since, on the average, more than 50% of manufacturing costs is assumed to be incurred by the assembly process, great efforts have been made to solve the difficult problem of mechanizing this process. So far only mass production oriented enterprises such as the automotive industry have been able to justify the expenses of mechanizing the process.

Note that the automatic assembly process imposes enhanced quality control of parts. The reason is that out-of-specification or faulty parts are likely to cause severe disturbances to the assembly process.

Real-time sensor-based capabilities in the computer systems are essential in the "move" environment. Material handling calls for plant floor terminals capable of two-way communication.

Test: We distinguish between inspection and functional testing. The variety of tests and checks that are actually made today is very great and will increase as the requirements for better quality and variety of products grow. "Test" is a subset of "verify," which is concerned with proper operation of processes and quality of materials, intermediate and final components and assemblies, etc.

Dimensional checking of production parts, chemical analysis, and electrical measurements on components and assemblies call for specialized equipment of the most diverse nature. In recent years a number of coordinate measuring machines have appeared on the market. Initially, the measuring probes had to be moved by hand while the coordinates were displayed or printed. The obvious next step was to use a small computer to control the machine and to interpret the readings. Such systems are being used today and will certainly be included as components of future manufacturing systems. The natural approach seems to be to add measuring capabilities, e.g., by attaching inspection machines to on-line machine control systems.

On-line chemical analysis will be performed (as it is being done today) by special instruments operated by computers. This form of testing is particularly important in continuous processing.

Electrical testing is singularly well suited for computer automation because of the limited amount of tailoring required to provide interfaces and the inherent flexibility of the stored program computer. In addition to measuring the signals generated by the test object, the computer provides the test sequence and controls the application of signals to the device under test. All decisions can be programmed into the computer, which also stores specifications and reference values. Such test systems are easily integrated into multilayer system structures for direct interaction with the various information systems.

The advent of computerized testing brings increased sophistication and superior consistency in quality control, substantially increased throughput and usually large labor savings. Advanced testing systems complemented with computer-controlled actuators will open the way to automatic adjustment, which is another step toward total automation.

Much analysis and interpretation work can be advantageously performed centrally (i.e., not on the plant floor) where considerable computation and storage capability will be concentrated. Again, efficient interaction between the equipment control systems and the information systems will be mandatory.

• *Implementation methods*

Before entering the stage of discussing specific technical requirements, some attention should be given to the question of how the integrated manufacturing control systems will evolve from the user's initial installation. There are many ways in which this can happen.

A user may have a "classical" data processing center which he expands to take up a variety of new tasks associated with his production problems. He would probably start with processing batch-type jobs concerned with production planning and scheduling. As the staff acquires proficiency in handling these jobs, the plant will become dependent on the output from the new planning and control tool. The next stage may involve shortening the vertical communication lines and introduction of the functions dynamic manufacturing planning and control. To this end, the user would have to implement a communications system including operator terminals on the plant floor. Once this subsystem is operating to satisfaction, we would expect the user to venture into the area of equipment control systems proper. The final stages would involve interconnection of the subsystems and integration of all functions in the factory. Such a job is never completed because there will always be new useful tasks that can be added. The evolution process described here is characterized by extensions of the system structure in both vertical and horizontal direction.

The time and investment required to attain the level of a properly operating integrated system are considerable; five years seems like an almost unattainable lower limit and periods of seven to twelve years seem more realistic. Important from a systems architecture standpoint, however, is the fact that the process is likely to extend over more than one computer generation, which imposes stringent compatibility requirements with respect to both hardware and programming.

A second mode of system evolution can be represented by the user who begins by implementing an equipment control system, for testing and on-line data collection. Assuming that the initial venture results in success, we would expect the user to proceed toward adding new tasks of similar type (i.e., horizontal expansion) before establishing a higher level information system to handle planning and operations control functions. Again, the systems architect must pay great attention to the problems associated with an evolution process which stretches over several generations of computer systems.

There will be many deviations from the basic evolution patterns described. Of essence is a realization that the system architecture must be flexible enough to accommodate practically any starting point and any form of system growth without forcing the users to invest in redevelopment work.

Because of recurrent changes of the application and the business environment in general, as well as the fast progress of technology, the objectives set for a new computer system installation will very soon have lost validity (often before the first implementation stage has been completed). Thus, the evolution of a user installation turns out to be a purposive and not a deterministic process, and the generally accepted design rules, which are based on a deterministic design approach, are not very useful. This climate of change contributes to slowing down the plant automation process and imposes drastic requirements for flexibility with respect to system design.

The previous discussion was mainly concerned with describing categories of tasks to which future manufacturing systems will be applied. To analyze system requirements in detail, we have to review also the ways in which the systems will be utilized. In this respect, we are likely to meet with revolutionary rather than evolutionary developments. The prime reason is that the manufacturing systems of the future will support a large portion of the people working in a plant and consequently have to be shared facilities. Efficient sharing can only be accomplished if a system has a high level of adaptability that allows it to look to each individual as if he were the sole user of the facility at any moment. In other words, a conversational interactive mode of operation will in due time dominate the picture.

As a consequence of this (and other factors) the activities accounted for in the previous survey will undergo drastic changes as the information technology advances. Thus, entire blocks of jobs considered important today will become irrelevant to the operation of the plant and will disappear. Their place will be taken by new tasks based on better information on plant status, performance and goals and on new approaches to plant control. Since the flexibility of the computer systems which conveniently permits restructuring of the job pattern is not necessarily matched by the capability of the work force to convert, major problems of reeducation and training are anticipated. This will impose needs for new system capabilities associated with the role of the facility as a teaching device and as an experimental research tool. These are considerations of great importance to the system architect.

System requirements

We are now ready to discuss detailed requirements, emphasizing those that have system architecture implications.

• *System architecture ground rules*

Our emphasis on the total systems approach and feasibility of efficient integration has strong impact on the architectural requirements. Some major requirements have been identified:

1) Data format compatibility through all levels of the total system. This may include standardization of data-set structures in the files. (This rule has to be adhered to with reason. If data are going to be generated, processed, and stored within a particular subsystem only, the data structure should be optimized for this specific environment.)
2) Programming language compatibility through all levels. (It may be necessary to deviate from this principle on the lowest levels of the control hierarchy where we may find engineers programming applications in special, high-level languages.)
3) Consistency of nomenclature through all levels.
4) Efficient channel-to-channel in-plant CPU-to-CPU communication.
5) Efficient communication between computers and terminals and between computers at different plants.
6) Possibility to operate subsystems on lower levels independently to insure fallback or to provide a minimum entry system which can grow into a total system.
7) More complete integration of sensor-based systems with operator-oriented systems at low hierarchical levels than is possible at present, i.e., within subsystems capable of operating independently. Penalities for providing this capability must be minimized to insure availability of truly low-cost starter systems.
8) Harmonious integration of computers of different vintage into a total system solution just as convenient as bringing together a structure in which all of the components are of the same generation. (We cannot tolerate a generation gap!)

• *System functions*

Processors

Factory automation requires the entire spectrum of processor capability, extending from the small dedicated binary processor to the large business computer. The real-time requirements increase in general as we move down through the hierarchy to the equipment control levels. Operation in real time calls for sophisticated interrupt schemes, clocks, interval timers, storage protect, automatic recovery, etc.

The capability to isolate functions, a combined hardware and software problem, is an important asset in the plant-floor environment where often development work under way involves the adding of new applications, the changing of a process, etc. Thus, the availability of 24 levels of interrupt in the IBM 1800 system has proven most useful, allowing practically every device to be associated with its own interrupt-level. In this way interrupts can immediately be identified with specific functions and devices and the operating system can be by-passed (with no analysis of what happened). As a note of caution, we should point out that while spreading the functions over several levels achieves a degree of isolation, that isolation is paid for by a decrease in throughput due to the preemptive nature of a priority interrupt structure. Proper isolation permits phasing-in new system components and applications with minimum impact on overall operation.

Adaptability to a varied and varying application environment calls for a high degree of modularity and expandability of the processors in the field. Adaptability to hostile physical environment is sometimes required. This is particularly the case with "intelligent" terminals.

The CPU's should be designed with extensive hardware checking of hardware errors and provide suitable interfaces to form duplexed systems.

Communication

The increasing dependence of computer support throughout the factory calls for the use of terminals to fulfill remote input/output needs. The spectrum of direct-line and common-carrier connections covers transmission speeds from 15 baud to 10 megabaud and the trend indicates future needs for even higher speeds. High reliability of the communication medium is required. Sophisticated multiplexors, possibly with stand-alone intelligence (stored program, read-only memory, etc.), will be required. Communication needs may be best met with a hierarchy of data transmission and multiplexing subsystems such as a broad-band coaxial cable network for CPU-to-CPU communication, a grid of twisted pairs of wire for slow and simple data-entry devices, etc.

Sensor input/output

The primary information path is a physical connection to the user's process instrumentation via the process input/output units. The wide spectrum of applications calls for adaptation to an extremely broad range of signal characteristics. Remote signal sources or receivers should be interfaced via remote sensor input/output units connected to the processors via high-speed communication lines. Room and power must be made available within the process input/output units to accommodate customer-built interfaces. The vendors should offer logic modules to ease matching. Major problems associated with sensor input/output are related to noise pickup, dc isolation, grounding problems, etc.

Man-machine interface

The need for improved management control of plant operations emphasizes and broadens the requirements for efficient man-machine communication. There is a need to develop techniques that allow customization without incurring special engineering and programming cost or creating unique manufacturing assemblying difficulties. The reliability of operator input/output devices must be improved and environmental limits extended. Transaction-based applications in the manufacturing industries will accentuate the need for low-cost graphics terminals of various kinds.

Interface into equipment

With the extension of the application base, there will be an increasing need for a standard digital interface. This is best illustrated by interfacing problems encountered in connection with computer control of machine tools and analytical instruments.

High capacity files

Manufacturing systems will have a need for large data banks from which information can be retrieved almost instantly. We see nothing unique in this requirement. The organization of the data bank must be studied in considerable detail. Techniques in how to sectionalize huge data volumes to optimize the trade-off between capacity and retrieval speed must be developed based on thorough knowledge of how much and what type of information is needed on the different levels of the organizational hierarchy.

In addition to a main central storage facility, some limited filing capacity will sometimes be required on the subsystem levels of the computer system hierarchy for the purpose of:

- Keeping the activity going on the plant floor, e.g., the operation of machine tools, when the central facility fails.
- Reducing the load on the communication system by recording historical data near the source.
- Buffering bursts of data which can not be transmitted in real time due to bandwidth limitations.
- Serving as data-base for a small starter system.

It is important that adequate measures be provided for protecting the files against errors due to malfunctions and against unauthorized access to confidential or otherwise restricted information. Data security and file protection are largely programming matters.

One technical implication of providing for files on the plant floor is the need to extend the range of physical environment in which such files may have to operate.

Programming

The computer programs, particularly those developed for operation on the lower level of a computer system hierarchy, must operate in real time. The factory automation environment can vary drastically from plant to plant and from the normal batch environment. Factory automation may imply 24-hour operation. This in turn implies that system maintenance and the debugging of user programs must be done while the manufacturing control system is operating on-line. Also, such a system must be able to quickly respond and adapt to changes in the environment; therefore, plant automation systems place partly unique requirements on the programming support that, in general, are not found in non-real-time systems.

The programming requirements fall into five categories:

1) high system utilization,
2) priority preemptive task switching,
3) problem/user oriented high-level languages,
4) communications based,
5) high availability.

Real-time factory automation programs tend to be input/output oriented. Also, programs in the set required to control any given process in the factory tend to be executed asynchronously even though they are related to and communicate with each other. The natural method of gaining high system utilization when a large number of asynchronous input/output oriented programs are to be executed is to run those programs under a multiprogramming system. Therefore, the basic operating system for a plant floor environment must be a multiprogramming system where the multiprogramming is controlled on an input/output delay time basis. Also, certain applications such as multiple test-stand control may require a time slicing algorithm within the system. Therefore, a generalized plant control operating system must provide an option for time-slice control of multiprogramming.

Batch operating systems are normally optimized for throughput. In order to achieve the optimized throughput, batched systems use a nonpreemptive task switching algorithm. (By "non-preemptive" we mean allowing a task to complete or suspend itself because of an input/output operation before executing any other task.) By using such an algorithm, the number of task switches is minimized and therefore the amount of system overhead is minimized. While such algorithms do increase the throughput of a system, they also decrease the responsiveness of the system to the outside world. Therefore, real-time systems require that the operating system use a preemptive task-switching algorithm. The priority of various tasks within the system determines the nature of preemptiveness between those tasks. Normally a priority preemptive task-switching algorithm is implemented via a priority interrupt architecture. A priority preemptive task switching algorithm will ensure that the highest priority active task within the system at any given instant of time will be in execution. Since such an algorithm will increase the number of task-switches within a given unit of time, the throughput of the system will be decreased. This is the price paid for improved responsiveness.

Another requirement placed on an operating system's task control concerns the manner in which a task may be initiated. Normally, batched systems only allow a task switch to be initiated through a JCL statement (Job Control Language). Real-time systems require that three other sources of job initiation be included in the operating system. The first of these three additional sources is the occurrence of external events. The priority interrupt which is triggered by an external event must be able to initiate a task the same way as a JCL statement does. The second additional task initiation source is a time-of-day clock. The user requires the ability to be able to specify to the operating system that a given task be executed in some relationship to the time of day. This may mean that he wants the task executed every hour on the hour or once a day at 8:00 a.m. Third, a task must be able to initiate the execution of some other task. This last source of task initiation means that the user, when writing in any of the acceptable programming languages within the system, can call the supervisor program at any point within his task to initiate some other task.

The third basic programming requirement of manufacturing control systems is for problem/user oriented high-level languages. In the factory environment, the engineer and technician on the factory floor will actually be doing the programming of their own specific jobs. Therefore, they will require an extremely high-level problem-oriented language to work with. A prime example of the level of language required is PROSPRO on the IBM 1800 system. One other characteristic of the plant floor environment is that the machines and applications will vary drastically from installation to installation and will even vary within one plant and with time. Therefore, it is necessary that the high-level languages be extendible and modifiable in their syntax by the user. This in turn will allow the user to adapt the languages to the specific manufacturing machines he is working with.

The fourth basic programming requirement for manufacturing systems is the requirement for totally integrated communications support. Because of the hierarchial structure of plant automation systems, the success of those systems is very dependent upon the communications between various systems within the hierarchy. Therefore, the communications control capability must be totally integrated into the operating system and must be totally supported as well.

The fifth programming requirement is for highly reliable and available systems. The process will continue to function independent of whether or not the computer system is up; therefore, the burden of trying to stay abreast of the process falls on the computer. By implication, then, the prime consideration in designing reliability features for a real-time system is error recovery, not error analysis. Only after such time as the computer system has recovered from the error and is back in synchronization with the process may time be used for error analysis.

The basic rule for real-time error procedures is that the system must be device nondependent. This means that if one or more input/output devices go down, the system must be able to either back up or disconnect those devices. No unit device should be so critical to the system that the loss of that unit causes the system to fail. This includes the system residence device. If the system residence device goes down, the operating system must provide for those programs resident in main storage at the time of the device failure to continue operation and respond to external events.

Real-time systems have two basic requirements of input/output device error recovery procedures. The first requirement is that the error recovery procedures must not interfere with the responsiveness of the system to the external world. This means that the programming system must not run in an interrupt-disabled mode while executing an error recovery procedure for such a period of time that would cause the response to external events to be critically degraded. The second basic requirement is that the user have control over the error recovery procedures. This means two things. First, the user must be notified of the termination status of an input/output request on a per-call basis. If the request is not successfully handled, the user must be informed what the failure was. Second, the user must be able to specify what the recovery procedure for a failure is on a per-call basis. These requirements apply to all devices in a real-time system.

While the programming requirements for real-time systems appear to be rather stringent in certain areas, various combinations of those requirements have been satisfied by various operating systems. An example of an operating system that satisfies all of the requirements today is the IBM MPX/1800 operating system.

• *Physical environment*

Plant floor applications sometimes call for operation on data partly available in a process environment. This environment is, in many cases, "hostile" and places particular requirements on the packaging of the equipment control subsystems. Sometimes complete satellite subsystems are expected to perform satisfactorily over the entire temperature and humidity range in which operators are expected to work.

Other factors may contribute to the hostility, such as dust, corrosive atmosphere, risk of explosion and mechanical impact, and strong electromagnetic fields. In the extreme cases, part of the equipment has to be mounted outdoors. Detailed realistic specifications related to the physical environment are difficult to develop. In general, the users cannot indicate critical limits or what they are prepared to pay for specific extensions of operating limits. Today vendor specifications are often set by what major competitors offer. An exception is the defense contract business in which the vendors are often requested to bid solutions to widely extended and well-defined requirements. Extended environment capability must be made available as optional features to minimize penalties on users not dependent on such quality.

• *Systems performance*

Throughput, system responsiveness and real-time behavior, etc., have been touched upon in various other contexts of this paper so we shall focus here on some less "glamorous" sides of system performance.

Realiability of systems and units

The most stringent requirements are related to ensuring continuous operation of facilities which are shared by many users and to providing continuous on-line control of a process or group of processes 24 hours per day for years. Of great economic and technical consequence is the fact that, although 100% availability is sometimes required, a substantial degradation of operating capacity (resulting from a malfunction) can, in many cases, be tolerated. Availability is largely a matter of systems organization and programming, and not just restricted to hardware.

A basis for achieving high availability and reliability is the provision of logic for extensive hardware error detection of hardware generated errors in all system components. In critical units such as the main memory and associated control logic error correcting circuitry may be called for. Where a satellite architecture is implemented, emphasis on reliability must be placed on the shared central processor and the communications system.

The factory automation system solutions must be planned and designed to permit the forming of system structures capable of meeting the reliability and availability requirements of individual customers.

Diagnostic routines and special servicing aids

The 100% availability requirement increases the need for efficient automatic diagnostic routines and on-line servicing.

In communication-based systems involving processors, transmission equipment (possibly common-carrier property) and terminals, efficient diagnostic methods are particularly important. Here we have the problem of directing corrective actions to the proper location and of establishing responsibilities between parties such as the vendor, customer and common carrier.

The trend toward letting the system user perform some system maintenance, dictated by his need to reduce the duration of unscheduled interruptions, will further increase the demand for automatic diagnostic aids. In particular, methods for using the central computer systems to diagnose remote terminals need to be developed.

Serviceability and maintainability

The requirements can be summarized as follows:

1) development of techniques allowing for servicing and maintenance without interruption of system operation, and

2) reduction of total maintenance effort without interfering with proper operation of user installations.

The main service and maintenance problems are associated with real-time systems operating around the clock for years. Here, the systems will have to stay operating while faulty units are being serviced.

With the trend toward communication-based systems, service and maintenance has increased in complexity. At central locations, it will still be possible to provide adequate engineering service and suitable premises for maintenance work. At the remote terminals, the support will often be poor and service facilities unsatisfactory. The importance of some form of maintenance by the system user increases as the system becomes more geographically extended.

• *Priorities and guiding principles*

Because of the complexity of the system architecture task which includes reconciling many diverse and occasionally contradictory requirements, it is necessary to grade the importance of meeting specific needs. The process of assigning priorities is governed by the answers to questions such as:

• How important is it that some particular capability be provided?
• How much is a specific capability worth to a user?
• How generally needed is a particular function?

With such knowledge and a profound appreciation of the application and business environments, the system architect can begin to develop his guiding principles. Reviewing priorities from a manufacturing systems point-of-view, we would like to assign high priority to the following requirements:

• Real-time capability on the different levels of the computer system hierarchy that is commensurate with the activities in different areas of the factory.

The change toward conversational interaction between the users and the system as well as the direct interconnection of computers with the production machines renders this requirement mandatory.

• Architectural flexibility (and versatility) and modu-

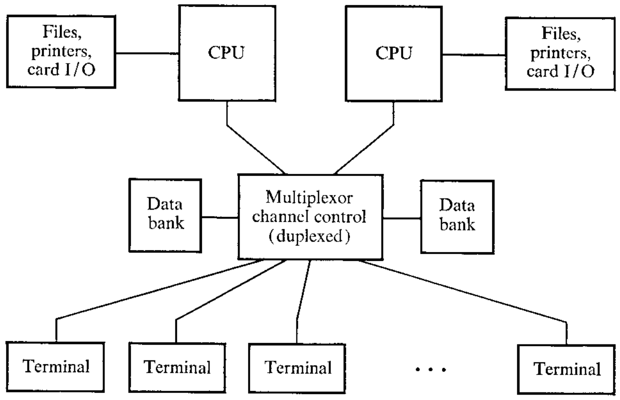


Figure 2 Configuration A—all computing functions centralized.

larity with respect to system functions, capabilities and power.

The changing of objectives that normally occurs during the evolution of a manufacturing system and the fact that we do not know beforehand what the system may be requested to perform promotes the subject requirement into the highest priority category. This includes facility to form computer system hierarchies of functionalized components and subsystems, and provision for alternative communication schemes to ensure satisfactory data exchange between subsystems.

• Capability to provide access to the computer systems at practically any physical location in the plant.

The particular access point can be a terminal or the control panel of a control computer. The requirement is a logical consequence of the assumption that future manufacturing systems will support a large portion of the plant personnel, who will often communicate with the system in an interactive mode.

The following guiding principles will be applied later in the evaluation of some basic system structures for implementing integrated manufacturing control systems:

• As a basic guideline, we propose that wherever extreme requirements have to be met, e.g., microsecond response to stimuli originating outside the computer, the capability should be restricted to the subsystem concerned with the particular problem calling for the exceptional function.

Thus, a large central computer, performing essentially management information system functions, should not necessarily be required to respond in microseconds, and an equipment control system should not be expected to maintain large data files efficiently.

The principle applies also to factors such as reliability and availability. The appropriate levels of performance must be provided within each segment of the total manufacturing system.

• The subsystems should allow implementation, expansion, and maintenance of individual applications with no impact on other activities. At the same time they should be capable of providing control to one or more machines or applications.

• Since many subsystems may be required to do the job in large plants, the architecture should enable centralization of various input/output units such as card equipment, high-speed printers, and disk files.

Centralizing and sharing relatively seldom used input/output equipment between many users have obviously a great cost incentive. The approach helps, furthermore, to facilitate maintenance and to keep paper and cards away from the plant floor.

• In addition to the input/output functions just mentioned, the central computers should provide all services that conveniently can be shared or for which large computing power and file capacity are essential.

This provision will render it possible to apply simple, low-cost subsystems on the plant floor and still provide adequate analysis capability to the operators. The central data banks will be used for storing application programs, logs, and much other pertinent information.

• The central computers should provide programming support to the subsystems on the plant floor.

Since the subsystems will often be very limited in capacity, jobs such as compilation, assembly, simulation, modeling and debugging can in many cases only be performed centrally.

Basic system structures

The system structure is a key architecture consideration. Possible variations in system structure are many indeed, so we shall limit the discussion to three basic structures.

• *All compution functions centralized—Configuration A* In this configuration a high-power central processing system is required to support the various areas of the plant via an extensive communications network and a variety of terminals. Here, the terminals are simple, they have no processing capability and, consequently, they are entirely dependent on the central facility. Because of this dependency, special measures have to be taken to ensure continuous service under all circumstances (occasional degradation of performance is usually acceptable), and duplexing of major shared components such as the multiplexor and the bulkstore is mandatory (see Fig. 2). A system structure such as this does not satisfy all of the high-priority requirements and does not reflect the thinking behind the guidelines. The basic rule is violated since:

• The central facility would have to provide a real-time capability satisfying the needs of the most demanding user in the plant. In a large general-purpose facility this may be very difficult and costly to accomplish.

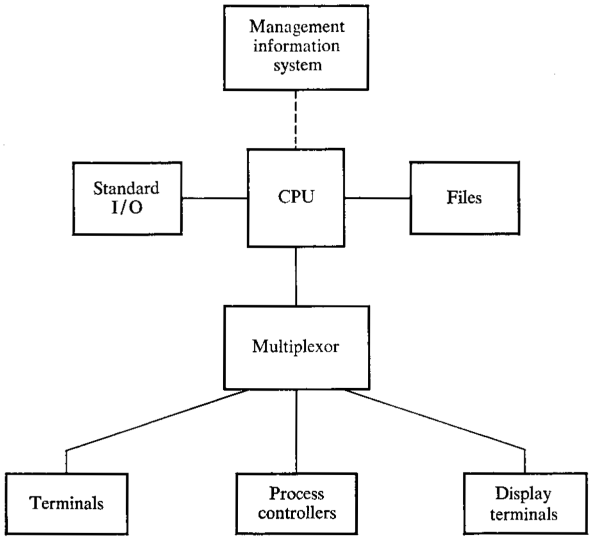
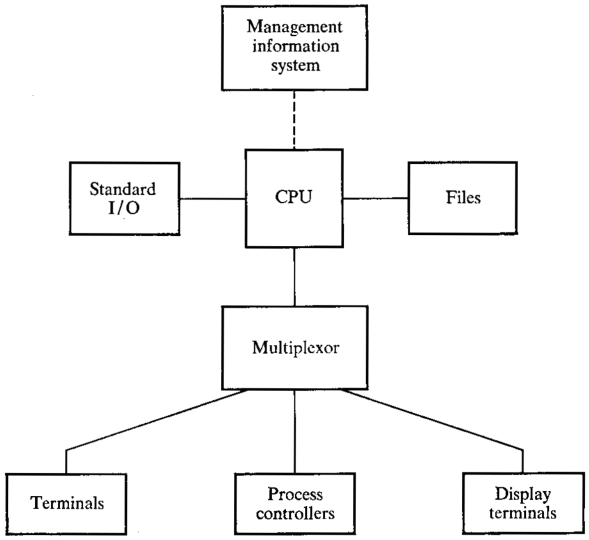


Figure 3 Configuration B—departmental real-time systems.



- A major portion of the system would have to be designed to meet extreme reliability and availability demands. This would lead to very high cost for equipment, system engineering and possibly installation.

Architectural flexibility was considered a critical requirement and, as far as processing capabilities are concerned, the structures would be quite satisfactory. They would, however, not provide an answer for the user who wants to start on a small scale. Tailoring to a specific application environment could cause some problems within this structural framework and the programming support would have to be quite sophisticated to keep engineers experimenting with new applications from disturbing operations in general. With respect to system growth, Configuration A may offer advantages.

Departmental real-time systems—Configuration B

Most tasks on the plant floor are handled in a time-sharing or multiprogramming mode by one or several expandible subsystems of relatively high power. One or more such subsystems may be connected to a corporate management information system. The IBM 1800 system fits very well into this pattern (see Fig. 3).

This structure has many advantages as compared to the preceding one:

- It provides considerable flexibility with respect to tasks that can be handled.
- It permits each subsystem to be operated as a self-contained stand-alone entity or as a component of an integrated manufacturing system.
- It allows for starting on a relatively small scale and provides growth capability, both horizontally and into multi-layer systems structures.
- It provides some capability with respect to ensuring high reliability and availability selectively, since support can be given to a critical application from two independent subsystems.

The structure also has some limitations:

- Since the subsystems are shared between many applications, the loss of one system may have quite far-reaching consequences and full duplexing of the shared components is a costly proposition.
- The processing power of the subsystem may in many cases be too limited for performing complex analysis tasks.

Two-level plant-floor control systems—Configuration C

This structure represents what is often referred to as the "satellite approach." It is based on applying a large number of more or less specialized or specializable subsystems to the jobs on the plant floor while coordinating the activities via a shared central computing facility (see Fig. 4).

The higher level system, a general purpose computer complex, provides a variety of services shared by the subsystems and the users on the plant floor. These services, which in general cannot be given by the specialized subsystems, include high-performance input/output functions (e.g., card reading/punching, high-speed printing), data-bank functions, program compilation or assembly and sophisticated analysis.

To make full use of the shared facility and bring without delays the services to every location on the plant floor,

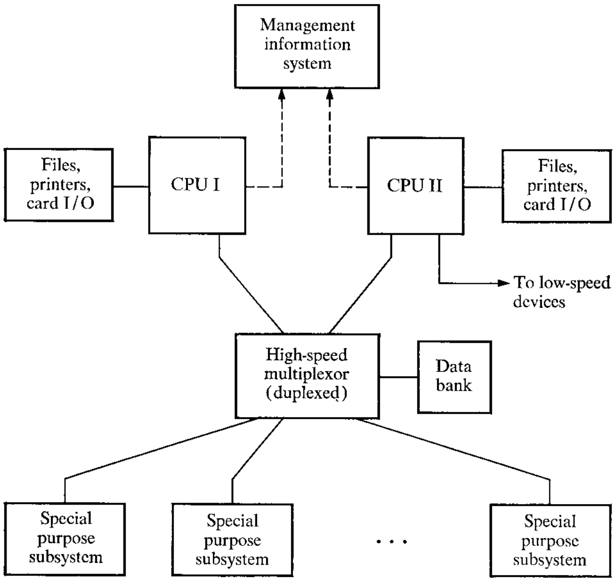


Figure 4 Configuration C—two-level plant-floor control system.

a wide-band communication system capable of transmitting data at about channel rates is needed. The multiplexing capacity should be expandible to accommodate hundreds of subsystems. A low-speed, low-cost network is also needed for connecting a large number of data entry devices such as work stations to the system. This second network may interface the system via a low-speed channel of CPU II or via a special-purpose subsystem operating as a line concentrator.

The central complex is duplexed to ensure continuity of service, but the two computers may be given different assignments. Thus, we may assign CPU I to handle the majority of real-time tasks, i.e. the handling of service requests from the plant floor. CPU II may be more or less dedicated to a variety of batch jobs which can be delayed should CPU I run into trouble. The subsystems are control systems capable of handling a wide spectrum of machine control and testing jobs.

The satellite approach offers practical solutions to almost all problems encountered in plant automation by providing essentially unlimited flexibility.

- It allows specialization of the subsystems to a level which ensures that specific tasks can be performed efficiently.
- It provides for full decoupling of system segments (subsystems) thereby satisfying the need for experimental facilities. These are needed for the development and phasing-in of new applications and changing of current tasks.
- The central resource can be applied to providing sophisticated programming support to the plant floor. In this way the need for professional programmers on the production line can be eliminated, and operators and engineers will write their own application programs using high-level problem-oriented languages.
- The two-level structure offers unlimited possibilities with respect to growth by allowing expansion in many different dimensions.
- High reliability performance on a selective basis can be conveniently realized by application of two subsystems to the same application.

The major disadvantage of the structure is that the initial investment may be high compared to what we would expect with systems of Configuration B. One practical (and economically attractive) way of implementing the two-level structure would be to begin with installing the central facility (as yet not duplexed) and apply it to various production planning jobs. With increasing experience the plant management would become more and more dependent on the services offered and, when convinced of the economic soundness of the project, proceed toward dynamic control of the plant by installing the first subsystem.

• *Comparison of basic structures*

When comparing the three structures, Configurations A, B and C, it seems that each may appeal to some user. Thus Configuration A may have some merit for a customer who has already invested in a good size computer installation, particularly if it is supporting remote terminals. The user may now decide to add such tasks from the plant floor which are not too critical with respect to system responsiveness or reliability. He may in fact include a large number of tasks.

Configuration B might appeal to a plant manager who wants to start automating on a limited scale and who is prepared to accept the fact that the subystems may run out of capacity. As a remedy, he has the option of adding more subsystems or extending capabilities by connecting to a higher-level system.

Configuration C, finally, is the structure preferable to the business executive who is really prepared to commit his enterprise to computer-based automation. Actual implementations of such structures and subsets of such structures are described elsewhere in this issue of the Journal. The structure can be attractive to smaller users as well if they start with the central facility and can make good general-purpose use of it while planning for the needs of the plant floor.

Relation to other complex systems

As we stated earlier in the paper, the systems architect is very anxious to find common features among the patterns of requirements generated in different application environments. Identification of important common elements allows him to emphasize these when writing architectural objectives. The desirable end objective is obviously a systems architecture which accommodates effectively as many key application areas as possible. We shall give a brief review of some application areas of importance that have requirements similar to those of manufacturing systems.

- *Data-base systems*

Ultimately, these systems will include as subsets a majority of file-oriented information systems. The key element of a data-base system is a central bulkstore containing all pertinent information related to the particular business operating the system. Access to the store is usually provided via terminals located in many places of the plant. An essential function of such systems is rapid retrieval of information according to some suitable search algorithm and transformation of the data into a form suitable for the requester. The transactions will in general be executed in conversational, interactive mode which implies good responsiveness of the systems and highly efficient search techniques. We expect most management information systems to evolve into data-base systems.

The management information system section of manufacturing systems is certainly no exception in this respect and consequently we expect many data-base systems to assume multi-layer structures forming computer system hierarchies.

- *Hospital information systems*

There are several areas within a hospital (particularly a large one) where computer systems are being used:

- General accounting and patient billing following the traditional pattern of batch processing.
- Control of analytical instruments in clinical laboratories involving real-time operation.
- Patient monitoring in intensive care wards. Here sensors attached to the critically ill patient are monitored on-line by a real-time oriented computer system.
- Nursing station guidance and information functions providing medicine ordering and scheduling, patient condition monitoring, and interdepartment communication.
- Support of various research activities involving both experimentation work, scientific batch processing, and data retrieval from large files.

These activities require a wide range of computer system functions. When integrating all tasks into a hospital information system, we detect many similarities to the process of integrating manufacturing-oriented tasks into one system. In summary, hospital information systems may advantageously be structured into hierarchies of interconnected computer systems capable of interacting with machines, instruments, and operators while running stacks of batch jobs simultaneously. Characteristic, also, is the need for extensive high-speed and low-speed communication systems and terminals.

- *Education systems*

Computer-aided instruction involves interaction in real time between a large number of individuals and a shared computer facility backed-up with large files. Depending on the structure of the institution that uses the system, i.e., whether it is a school, a school district, a university or a private industry, the education system will take the form of a more-or-less extended network of terminals, communication links, multiplexors, and computers possibly organized as a hierarchy.

A system architecture tailored to the needs of the manufacturing industries may very well provide a good base concept for an educational system.

Conclusions

Plant automation constitutes a major subset of a large family of applications calling for consolidation of data processing, various human mental tasks, equipment control, and testing into integrated hierarchical structures of computer systems.

The development of a flexible system architecture that can accommodate system components of many different kinds while giving sufficient freedom of choice with respect to structural alternatives is critically important to the progress of computer-based plant automation. A system architecture that focuses attention on the plant-floor control system would probably not be satisfactory since the integration of equipment control functions into the management information system oriented control functions is an essential part of automation.

The automation (or system integration) process will often stretch over time periods which are longer than the life span of individual computers (or subsystems). Thus, the system architecture must offer means not only for interconnecting computers of different functional orientation; it must, in fact, provide for integration of machines of different vintage as well. This problem is closely related to the architecture task of planning for system expansion and migration.

To satisfy the needs of the integrated manufacturing systems, the system architecture must accommodate at least one complete communication system.

The greatest challenge for the system architect is that he will have to create a framework within which system designers can structure manufacturing systems capable of handling besides the planned tasks a variety of unforeseen applications. We conclude from this and from the previously shown similarities between the requirements of several, in essence, unrelated application fields that a system architecture which really satisfies the needs of the manufacturing environment will be a useful frame for system design in other application fields.

References

1. J. E. Stuehler, "An Integrated Manufacturing Process Control System: Implementation in IBM Manufacturing," *IBM J. Res. Develop.* **14,** 605 (1970), this issue.
2. J. Calva, "PCOS: A Process Control Extension to Operating System/360," *IBM J. Res. Develop.* **14,** 620 (1970), this issue.
3. F. W. Thoburn, "A Transmission Control Unit for High-speed Computer-to-computer Communication," *IBM J. Res. Develop.* **14,** 614 (1970), this issue.
4. R. B. Fischer, et al., "Computer-aided Testing and Fabrication of Magnetic Tape Heads," *IBM J. Res. Develop.* **14,** 633 (1970), this issue.
5. M. R. Radio, et al., "Precision Automatic Measuring of X-Y Coordinates," *IBM J. Res. Develop.* **14,** 641 (1970), this issue.

Received February 4, 1970