Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

C.L. Blom, A. Choudry, P.J.W. ten Hagen, L.O. Hertzberger,
A. Janssen, A.A.M. Kuyk, F. Tuynman, W.E. van Waning

A strategy for computer integrated manufacturing systems
Processing and communication

Department of Computer Science Note CS-N8403 May

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

A STRATEGY FOR COMPUTER INTEGRATED MANUFACTURING SYSTEMS PROCESSING AND COMMUNICATION

C.L. BLOM, P.J.W. TEN HAGEN, A. JANSSEN, A.A.M. KUYK, W.E. VAN WANING
Centre for Mathematics and Computer Science, Amsterdam

A. CHOUDRY, L.O. HERTZBERGER, F. TUYNMAN
University of Amsterdam, Department of Computer Science

In 1983, the European Economic Commission launched a research program called 'European Strategic Program for Research and Development in Information Technology' (ESPRIT). Since manufacturing is an important application area of Information Technology, attention has to be paid to the evolving integration of Computer Aided Design (CAD), and Computer Aided Manufacturing (CAM). In order to understand and optimize this process, a research project was initiated to establish 'Design Rules for Computer Integrated Manufacturing (CIM) Systems'.
As a first step, the state-of-the-art in two main areas of Information Technology (Processing and Communication Technology) was investigated with respect to key modules of both CAD and CAM, in order to identify well-founded strategies aimed at optimal support for the various Computer Integrated Manufacturing activities, especially from the point of view of Processing and Communication Technology.

1980 MATHEMATICS SUBJECT CLASSIFICATION: Primary: 69L60.
Secondary: 69K21, 69K35, 69K64, 69C20.

KEY WORDS & PHRASES: Computer Integrated Manufacturing (CIM), CAD/CAM, Flexible Manufacturing Systems (FMS), sensor systems, robot vision, solid modeling, design, computer networks, processing technology, communication technology.

Note CS-N8403
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

0. Introduction to a processing and communication strategy for Computer Integrated Manufacturing.

The Information Technology for Computer Integrated Manufacturing (CIM) can be divided into three main subjects:

- Processing Technology
- Data handling Technology
- Communication Technology

For each of these subjects a strategy can be developed, which explains how it is to be applied in CIM from a technological point of view.

In part 2 a strategy is developed for two of the above subjects, being Processing and Communication.

The strategy chosen aims at optimal support for the various CIM activities, from the point of view of economics and flexibility. A further motivation for formulating these strategies is to be able to build general applicable CIM subsystems, with predictably good and reliable processing and communications characteristics.

A starting point for such a strategy is to formulate the processing and communication requirements for CIM, both general and specific to a CIM-module. These requirements next must be provided from the available processing and communication resources, build and programmed in the most appropriate way. The selection, now or in the future, of a certain method for processing and communication in a given CIM situation thus may depend on existing or forthcoming Information Technology.

It will turn out that for communication a more global strategy can be formulated. For processing, in many cases, a more detailed analysis of CIM subsystems is necessary in order to identify their characteristic processing requirements (processing characteristics).

Chapter 1 of part two explains how an analysis of CIM activities can be done in order to find processing and communication requirements independent from the way a CIM system is organized on a functional level.

Chapter two justifies why initially the two modules computer aided design and flexible manufacturing systems have been chosen. Later the strategies for other modules are developed. This is in accordance with the ESPRIT project schedule.

Chapter three outlines the method for obtaining a processing strategy.

Chapter four analyses design activities and formulates general processing characteristics of design support systems.

Chapter five treats flexible manufacturing in a similar way.

Chapter six analyses a key module of CAD/CAE, being geometric modelling and graphics, in detail. As a key module, it determines to a large extent the processing characteristics of CAD/CAE as a whole.

Chapter seven treats sensor systems in detail. Their processing characteristics are going to be of major influence on FMS systems processing.

Chapter eight introduces a communication strategy and explains its more global nature.

Chapter nine gives an overview of the state of the art in communication facilities and highlights those aspect especially relevant for CIM.

Chapter ten, finally, puts current strategies in the perspective of developing and improving information technology. This perspective is based on actual observations, as the one given in appendix A, rather than a speculative analysis.

Appendix A is a report of a study trip to Japan, made as part of the ESPRIT project.

1. A Reference Model for CIM

1.1. Computer Integrated Manufacturing

A CIM (Computer Integrated Manufacturing) system is a configuration of cooperating computer systems and computer peripherals for the complete control of a manufacturing process. The CIM system contains computer systems for designing products and production lines, planning of products and production, control of production and storage of products.

There is also a planning and scheduling subsystem required for making all these facilities work well together. This aspect will for the time being not be covered.

In a completely computer integrated manufacturing system all components are working together directly, without need for human intervention. For instance, a completely designed component, will have all the specified information stored in the CAD subsystem. This allows for automatically generating as well as installing programs for manufacturing, and subsequently ordering/scheduling production.

In practical situations CIM will be more or less incomplete, and human intervention will be essential for it. However, the reference model applies to both, complete and incomplete, situations.

Because the processing strategy to be developed will focus initially on two modules, one in the design area and one in the manufacturing area, (i.e. CAD and FMS) the model will be defined only as far as is needed to encompass these two modules.

A full modularization of CIM functionality will not be given here. The computing support given to CIM activities has, in practice, lead to a number of established computer subsystems, such as drafting systems, finite element analysis systems, a robot simulation system or a crane control system.

It is to some extent possible to consider those support modules and analyse their processing needs independent of the way they are grouped together. This grouping and mutual cooperation method depends on the true modules defined for CIM.

The rough division in CIM modules used here is already mentioned in the initial study proposal. Using this division, we immediately step down to the processing level to identify CIM computer processes and their processing characteristics.

The overall CIM system breaks down in two obvious subsystems: Design (CAD) and Manufacture (CAM). For every activity in manufacturing, there is a corresponding design phase. The integration of the whole system has a number of axes.

1. Integration between design and actual use.
2. Integration between various design activities.
3. Integration between various manufacturing activities.

Integration can be achieved by more than one path resulting from movements along these axes. For instance: manufacturing constraints of a product that influence its design, may be obtained through a number of possible integration channels. The manufacturing constraints may be obtained by the CAD system from the computer aided production engineering (CAPE) system, or directly from the actual status of the manufacturing system.

So the top of the module tree looks as follows:

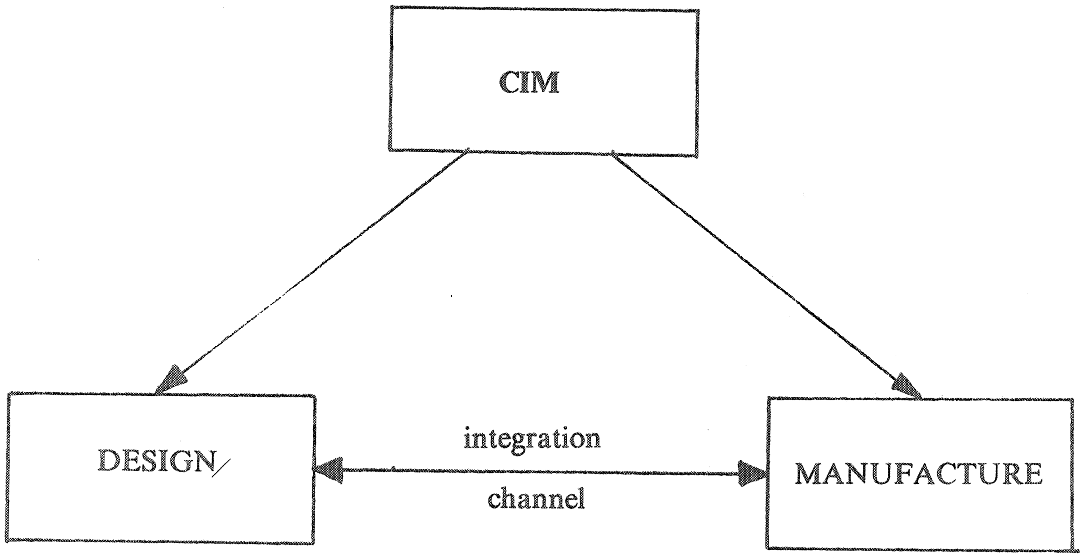

Figure 1.1

1.2. Design

In an integrated system there must be an overall design strategy. This strategy determines how the complete design activity in a CIM-system is organized. For instance, the strategy must give rules how the information system for design should be organized as a number of cooperating databases (knowledge bases). It must decide which information is stored for retrieval as opposed to generated, prior to use.

From this strategy it follows how many different designers can be working on the system on related topics.

The calculation processes carried out during design must be classified primarily in on-line and batch processes. Only the former can be used in an interactive design system. Design productivity increases dramatically when computer aided design can be done interactively. The processing characteristics of the design calculations must be defined in order to make such a classification.

1.2.1. Design Activities

Following the primary subdivision we will divide the design activity further. First of all there is the subdivision in design of products and design of production methods.

It is obvious that integration requires the transfer of product design information to production design. For instance, the shape information may be used to generate NC tapes. Also the reverse path is required, in order to give the designer suggestions for product architecture which favors existing production methods.

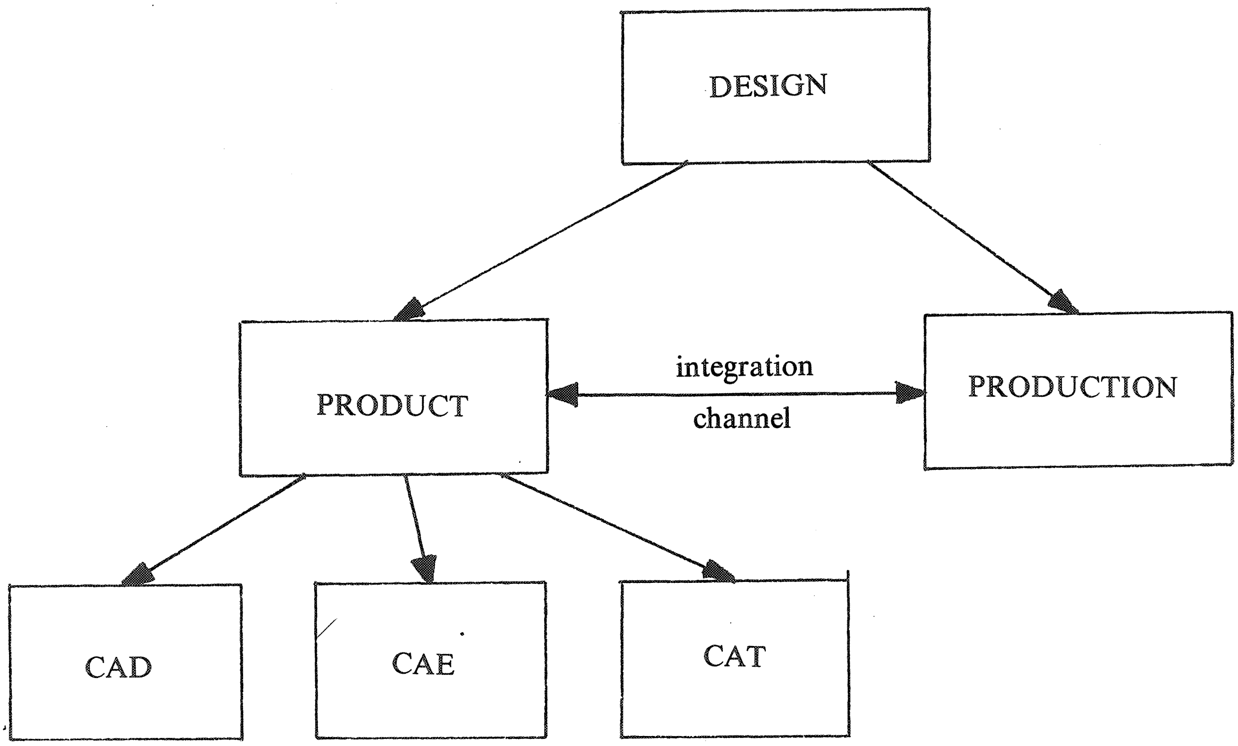

Figure 1.2

1.2.2. Product design

The three major modules of product design are:

CAD, computer aided design, all functions to assist a designer in defining his product, with the effect that the final definition will be represented in the computer. CAD uses several basic functions such as drafting, shape definition, materials selection, etc.

The modules CAE, computer aided engineering, and CAT, computer aided testing, are for analysing the design as produced by the CAD system. The CAE module uses the design information for generating new representations for the purpose of calculating relevant physical properties (e.g. strength) and to simulate product behaviour.

The CAT-module is a system for testing of actual products built (prototypes). It contains test sequence programming, controlling the test experiment, data collection and analysis.

CAD, CAE and CAT can be further subdivised in functional modules as indicated in fig. 1.3. In turn when considering computer support for these modules (e.g. shape definition, finite element method) there is a connection with existing computer systems dedicated to give this kind of assistance to the designer. Equally important is the computer process which makes the various types of assistance work together well. For this some knowledge about design methods in general is necessary.

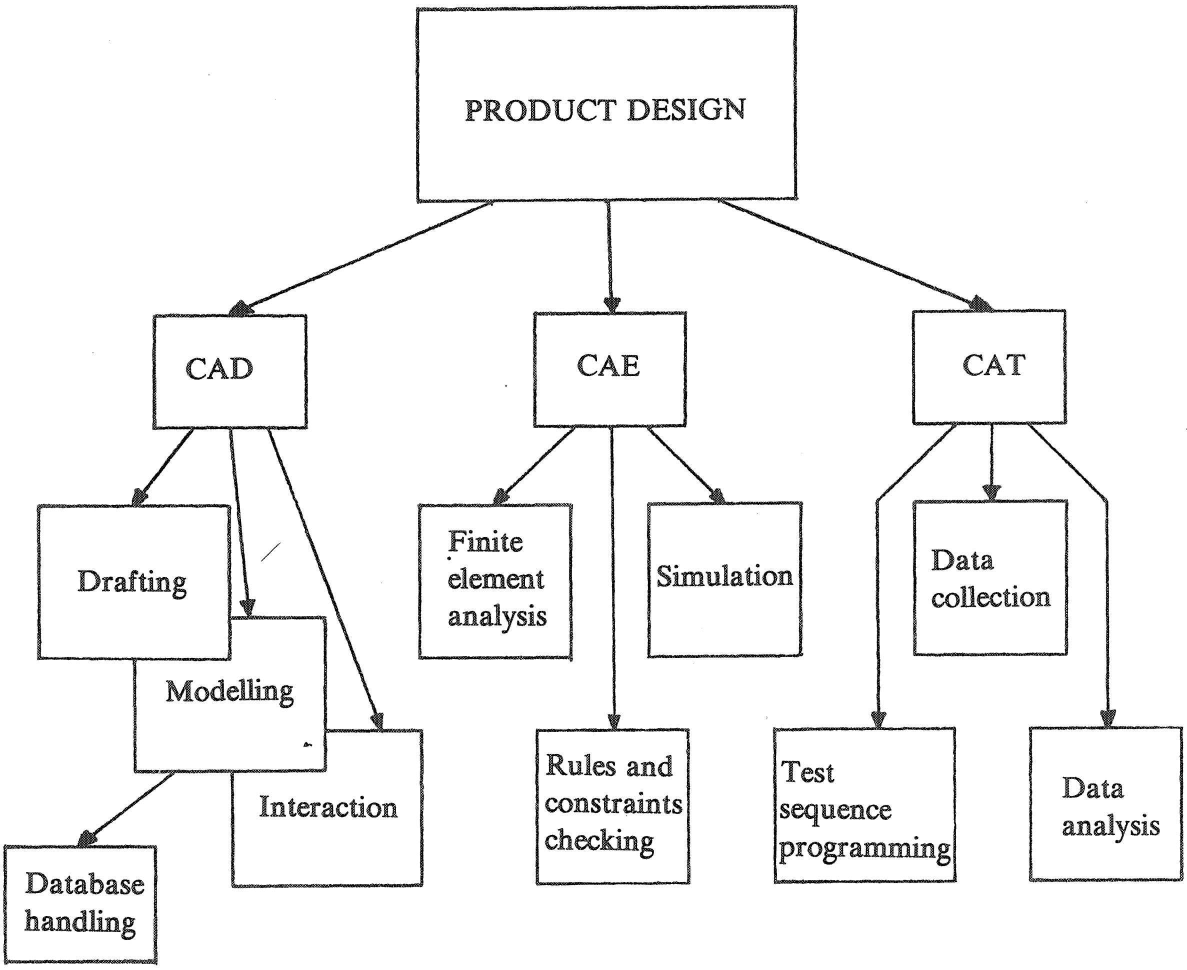

Figure 1.3

1.2.3. Production engineering

Computer aided production engineering is the process of defining completely how a product is to be manufactured, given as input the product specification and the manufacturing means.

The production engineer is situated between product design and actual manufacture. In an integrated system he will directly use both product specifications and manufacturing data. In many cases he will also mediate as production expert for the designer on manufacturing constraints.

Computer aided production engineering has many subdisciplines which in turn are supported by many modules for which computer assistance exists as a computing subsystem. Important modules are the following:

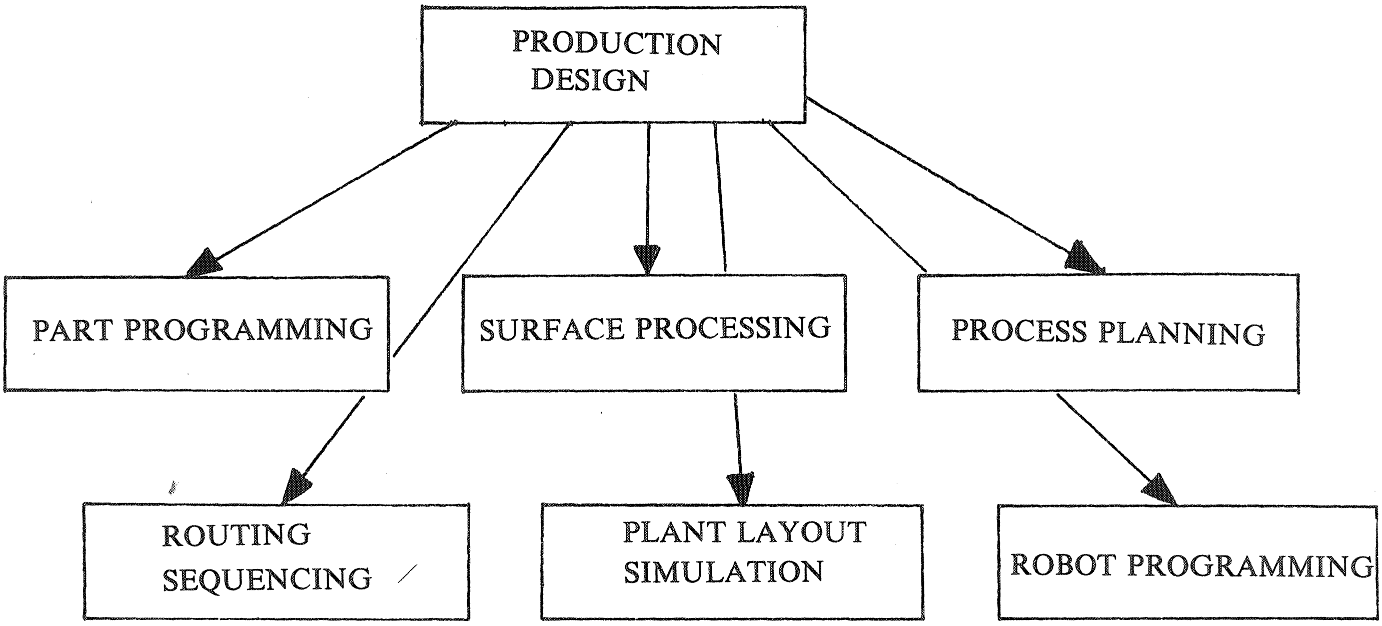

Figure 1.4

- *Process planning:* The task of determining how the manufacturing of a product will be split in smaller tasks which can be assigned to a manufacturing facility. This task definition and assignment will be done both for machining and assembly. Important input also is the knowledge about subcomponents which are already available.
- *Part programming:* The task of specifying how a (basic) part is to be machined. This may lead to generation of NC-tapes etc. Part programming becomes very important when in an FMS environment a component strategy allows for many parts to be defined as variants of the same basic component.
- *Surface design for processing:* This special case of part programming is treated separately because the design and processing design of complex surfaces very often leads to adjusting the design. In addition 3D cutting paths for complex surfaces require special attention.
- *Routing and sequencing:* The task of determining machines and assembly sequencing and assignment of tasks to specific manufacturing cells on the shop floor. Here optimization techniques play an important role. Flexibility is very important in this case in order to supply alternatives in case of a machine breakdown.
- *Plant layout and simulation:* In an integrated system the available overall knowledge of the situation an process designed easily allows for simulation of a manufacturing process before implementing it.

- *Robotprogramming:* The task of generating instructions for robots. This can be at two levels: programming of a specific task, something which is currently done by teach-in, it can also consist of the activation of a preprogammed task. The latter can be centralized.

Each of the subsystems described must have an interface with the overall CAM-design controlling process. In addition many subsystems need to interface as well. Typically for design processes is that they provide processing instructions (and data) for other processes.

1.3. Manufacturing

Manufacturing has four major subjects, being:
The building of manufacturing machines, the programming of such machines for a specific task and the control of such machines when in operation. Last but not least, there is a subject dealing with the overall control of the manufacturing process: synchronization and cooperation of the various manufacturing subsystems, materials support between them, etc.. For each of the manufacturing processes a corresponding design activity can be identified. All these are cases of production design (cf. 1.2.2).

Rather than subdividing further along the four lines given above, attention is given to a second independent classification, being the manufacturing type. The overall control system is still maintained, because, to some extent, it has to be independent of the manufacturing type. For each type the remaining aspects are being considered (i.e. build, program, control).

The important manufacturing types (from the point of view of integration) are: controlled machining, controlled assembly, flexible machining, flexible assembly, robotics and transportation. Again this is an incomplete list, but it gives sufficient breadth for putting flexible manufacturing in perspective.

In accordance with the above one can classify manufacturing activities as follows:

The controlled machining is the already existing, technology of providing a machine tool or an assembly machine with a series of data (e.g. NC-data or component selection and positioning data), which will be interpreted by a fixed program. There is no feedback from the machine tool about the result. The supply of parts and tools has to be done by hand, as has the removal of parts, tools and waist material. Certainly no robots are applied. Nevertheless, this restricted use of computers as converters of machining specifications into servo control instruction has brought a great improvement of productivity.
In a similar way one may view isolated robots, such as welding robots which are installed, more or less on an ad hoc basis, in a production line, sometimes more because of their better predicate quality output.

The borderline between controlled manufacturing and flexible manufacturing is very vague. Flexibility can vary continuously between totally unflexible and totally flexible. Flexibility can be achieved by combining a number of means, each with its own contribution to total flexibility.
Intuitively a flexible manufacturing system is built by combining a number of flexible manufacturing units or cells. The flexibility is further increased by allowing flexibility in the combinations possible for the various units. This means, for instance, that a flexible

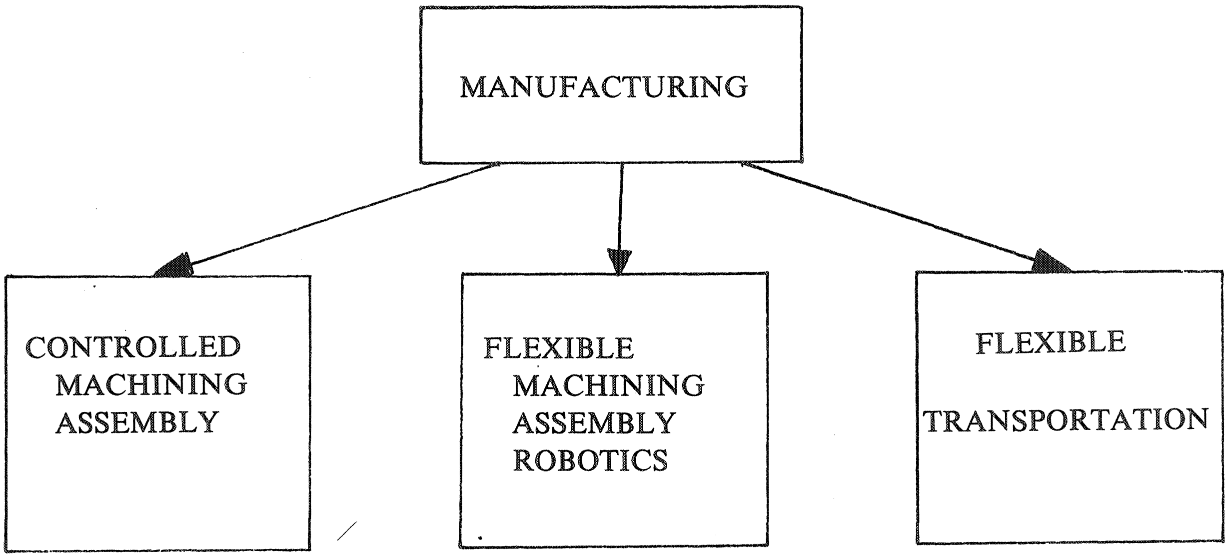

Figure 1.5

transportation system can make a considerable contribution to the flexibility of the system as a whole.
The flexibility of the manufacturing units stems from several properties: they can hold a variety of tools, they can handle a wide range of parts sizes, they have many degrees of freedom in positioning part and tools. They are driven by a reprogrammable computer system. They can be remotely controlled and connected with other systems such as a transportation system. Last but not least they can monitor their own status and production results through built-in sensor systems, which are also under computer control. A flexible unit also can communicate with the central controlling unit or with other units or robots.

The basic processing modules for FMS are not that well established as are many for CAD and CAE. Flexibility is only recently recognized as a key to economic manufacturing. The concept of flexibility nevertheless leads to a number of machining -, assembly - and transportation architectures supported by a large variety of robot tasks, which, dependent on robot flexibility, can be handled by a relatively small robot family.

The overall picture currently is that each of the three aspects (machining, assembly and transportation) is realized through fairly powerful units which can be combined in arbitrary configurations, because each unit can be combined with arbitrary other units.

The basic processing support can, based on this analysis, be categorized and characterized. This will be done in chapters 6 and 7.

2. Representative CIM modules

The CIM modules treated in this study are chosen in such a way that most of the characteristics of the integration problem of computer aided design and manufacture can be touched upon.

One of the modules chosen is Product Design and Engineering. This is because there is a relatively long experience with computing support for these activities. The processing and communication strategy for design can be analysed using existing examples. Also the market offers a wide variety of design and analysis support applications. For this area processing and communication must be seen as existing (established) resources, which constantly are subjected to an evolutionary process due to introduction of new Information Technologies.

Another important aspect of design in the realm of CIM is, that the total design work for CIM (product, production, production methods) reflects the complete CIM activity range. Therefore on the fairly abstract level of design methodology it is necessary to find already the principles for the architecture of the design systems. The design activity must be organized in such a way, that the integration of the three components, mentioned above, is reflected in the architecture. The total design activity in the integrated environment will be so complex that only the most carefully chosen architecture of the system will be able to give the right kind of support. Much of this support will be in guiding a designer through a mass of information to the relevant pieces.

Still on the lower level of assistance for individual design tasks, the various support modules will be found in much the same way as for a classical uncomplicated situation. It will still be the case that some of the modules providing assistance will have far greater influence on the total processing and communications characteristics than others. The more complex integrated environment will, in a straightforward manner, lead to equally more complex low level support modules. The increase in complexity of such a key module with increasing integration may greatly change its processing and communications characteristics.

The most important of these modules in the design area is the geometric modeller. The computerized models of products, production methods and production devices all are built around geometric models describing their shape, spatial properties and (de)composition. Integration begins at being able to geometrically transfer model components to the same world for confrontation. It is for these reasons that within the design module a geometric modelling module is chosen for further detailing design rules for processing.

As far as communication is concerned, the strategy design rules will remain on a more global level, because a CIM system very likely will be linked up by a small number of typical communication systems. The characterization which will allow for selecting a communication method will be done mainly on the more global design level. A more detailed analysis may be necessary when a communication protocol strategy is to be selected.

The second major module chosen is Flexible Manufacturing Systems. In contrast with design, flexible manufacturing is a relatively new area, where apart from the basic goals, little has been established. What has been achieved in this area so far, shows great potential for improving both quality and economics of manufacturing.

Currently, the major emphasis in FMS is on defining a computer system for controlling the FMS-hardware (i.e. the parametrized manufacturing method) in the optimal way. The computer system should fully utilize the cell as is. The FMS-hardware is only programmable before use. Next the computer system must try to exploit this task repertoire in an optimal way.

A basic computer system architecture for even this limited flexibility is not established yet. Following this, a whole range of programmable features must be added for increasing flexibility. For this much research is still to be done. The limitations for FMS components also have to do with the necessity to keep the complexity of the integrated system within bounds that can be handled with existing Information Technology.

Defining a processing and communications strategy for FMS will therefore be a design task rather than a synthesis task. As is the case with design, also with FMS a number of low level support modules can be identified. One key module for FMS is sensor systems. Sensor systems are as yet not completely understood. There is no established programming methodology and it is. an outspoken example of a module which will greatly change in the near future because of new developments. A processing strategy for sensor systems will therefore have to pay special attention to IT evolution. For flexibility combined with high reliability which is required for unmanned manufacturing units, sensor systems will be vital components. These processing characteristics may be relatively well separable from other FMS processing (e.g. calculating cutting paths). However, the integration requirements will necessitate that the sensor tasks can be controlled from outside the manufacturing cell. Thus the communication requirements will have to be given special attention.

The second module chosen therefore represents the other extreme in CIM-modules, being one still heavily under initial development.

3. Introduction to Processing Strategy.

Design rules for computer processing in CIM.

In this section it is defined which aspects are relevant for a computer processing strategy for CIM. The processing strategy is an IT strategy i.e. a systematic method of using Information Processing Technology in providing the best processing support for CIM modules. This strategy should lead to two classes of rules;

- how to build basic processing systems for CIM tasks (e.g. preferably by using combinations from a limited number of more basic tasks).
- how to organize the distribution of processing over a number of available processing systems given the requirements of the CIM situation.

For the first class of rules existing hardware and operating systems as well as very basic packages such as a graphics system need to be considered. The future trends in hardware and system software need to be taken into account in order to arrive at competitive components, or, implementations of basic components. Standard interfaces, functional or other must be defined in order to provide for flexible packaging.

Very important for this set also are the processing characteristics (e.g. speed, resources required, real-time, reliability) which need to be supported by such subsystems. The ability of subsystems to fulfill a given set of requirements for a module need to be quantifiable.

For the second class of rules the external control of processes is the major issue. It seems unavoidable that processes can be subjected to a common control language so that processes of different origin can be combined in one distributed environment.

Also the distribution problem itself for CIM is specific for this category. Much can be learned here from the process control area for non-discrete manufacturing. A distribution scheme for CIM processes has many levels each of which may use a different type of specification method, for instance, a sequence of processes may have a control structure totally different from a parallel set of processes operating on the same task.

Modules and Standards

A fully modularized CIM system can be subjected to an analysis to produce general purpose CIM functions (such as crane control, production sequencing).
Whenever an opportunity exists, such general purpose modules must be identified. Rather than working out in detail some of these basic standard module descriptions, as many candidate modules as possible must be found. They deserve special attention from the point of view of a processing strategy.

The modularization itself, especially its refinement can be influenced by the attempt to maximise the number of "standard" basic components. Special attention must be paid to the fact that such standards often can only result after reducing a module's task until a more general purpose but smaller task remains. Such task reduction must be justifiable by a new task structure which shows that no essential subtasks have disappeared.

The processing strategy rules next must make it possible to find one or more ways to produce a cost effective good quality process for such a task, either as a special purpose processor or as a subprogram scheduled for a more general purpose installation.

In order to identify similar processing requirements in various modules, a uniform checklist must be drafted, which is going to be the *core* of the processing strategy rules. Each module can next be characterized following the processing checklist.

The checklist will allow for each item a general "strategic" method to be followed or a specific one, particular to the module.

According to this analysis of design rules for CIM processing, the following four categories of design rules can be identified.

3.1. Processing Characteristics

In order to be able to formulate the processing requirements of CIM modules a set of terms and references is needed. Also in order to find out whether a given processing component can be used for a CIM-task, it must be possible to clearly define its supporting abilities. Both sets of terms will then be used to describe a processing component with abilities to fulfill the requirements of a CIM module. The two sets of terms are listed in a so-called list of *processing characteristics.*

An attempt will be made to keep the list short, in other words, to have only important characteristics on it. This is essential for the list to be useful. The characteristics list is given in fig. 3.1. In the sequel a short description of the items on the list will be given.

Processing types.

Applications can be categorized according to a computing resource or facility they predominantly use. CIM-support modules seen as application modules exhibit this property very clearly. CIM-modules considered on the level of detail where they can be associated with computer processes usually take care of one particular CIM task. This means that the processing characteristics are derived from that particular task only. Examples of processing types are given in the list (e.g. number crunching, real-time control, etc.). For many of these types dedicated systems (hardware-software combinations) exist. However, if they can be applied depends on additional characterisitcs.

Locality requirements.

The *locality requirements* determine to what extend a processer is to be tied to a certain place. This may be because that is where the data are, or because that is where the unit to be controlled (control sinc) is, etc.
The locality requirements put constraints on the freedom of choosing resources. Conflicting locality requirements for one process may even lead to further distribution.
Conversely limited resources may also lead to a certain type of locality requirements. However, this kind cannot be accounted for in design rules, because they do not follow from the CIM architecture, but from specific situations.

Reliability and recovery.

Both properties for CIM are extremely important. In fact many of the computer systems available on the market provide insufficient reliability and insufficient recovery support, although for other applications they can be considered adequate.

A Checklist for processing characteristics

Basic elements of processing

- CPU power
- memory size
- operating system requirements
- i/o handling
- expandability (without replacement)
- number of users
- number of tasks
- filing requirements
- communication requirements
- acounting

Locality requirements

- data base access
- control sinc
- instruction source
- monitoring requirements
- environmental conditions

Reliability and recovery

- requirements
- methods:
  - duplexing
  - standby
  - sharing
  - shedding
- back up
- restart
- mean-time between failure
- support for maintenance
- diagnostics
- mean-time to repair

Monitoring

- progress
- correctness
- man/machine interface
- logging

Processing types

- number crunching
- real-time control
- data handling
- interaction
- communication

Figure 3.1

3.2. Processing types assigned to CIM modules

While analyzing the various CIM activities it should become clear what functional units can be given computer-processing support. The first question to ask for each CIM-module is: what is the nature of support needed: e.g. doing calculations, issuing control, data handling, monitoring, etc. This first major qualification determines the processing type for this particular support. Next further CIM requirements need to be considered, in order to find out under what circumstances or in what environment this support is to be given, e.g. is it a real-time environment, what reliability is required, can it be installed remotely etc. Important is to only select the relevant characteristics. Finally this combination of characteristics should be further completed by indicating which combinations of basic elements of processing could adequately support it. This combination is not necessarily unique.

These assignments of the characteristics constitute a second set of processing strategy rules.

3.3. Methods for processing support

Support for a certain type of computing e.g. very accurate, very reliable, at the very right moment, etc., can generally speaking be given in more than one way, even for one specific type. Also the effectiveness of the support may depend on the application and other environmental conditions. This means that choosing a method for supplying sufficient support may become very important. It will in many cases greatly influence cost. Economics may force one into choosing a method which is not optimal in an individual case, but may be wider applicable. It is the process of defining the processing means that the processing characteristics of various modules will be compared and combined into processing characteristics for higher level modules.

Eventually based upon all this information a method has to be selected. Doing so will become a lot easier if a processing strategy could be based on relatively few support facility configurations. Finding these is sufficiently important, to justify changing processing characteristics or the modularization chosen. Nevertheless sufficient flexibility must remain to allow exceptions.

4. Engineering design viewed from an IT-perspective.

4.1. Introduction.

The purpose of engineering is the design and construction of a specified object. The specification of the object to be engineered may vary from a set of desiderata to a highly structured set of prescriptions not only containing technical material, but also demands from for instance economic and production-oriented areas.

An engineer analyses specifications the result of which include:

- an identification of parts from which the intended device is to be built;
- the structural interrelationships the identified parts are to have in order to realize the functions of the intended device;
- (sub-)specifications of the functions and constraints those parts have to satisfy to meet the specifications of the intended device.

Whether or not a design plan meets its set of demands and constraints will have to be verified. The identified parts have to be synthesized according to their function in such a way that the initial specification will be satisfied. Establishing the acceptability of a plan with respect to a specification can, in many areas of industrial engineering, be done by computational means: mathematical and physical models make it possible to capture essential aspects and to compute the parameters necessary to judge the acceptability of a plan.

Like any problem-solving activity, the activity of engineering can be subdivided into three aspects:

Specification - a (more or less) detailed description of a device to be constructed, constraints to be imposed on materials and workmanship needed to be undertaken by an engineer, architect, &c;

Analysis - the resolution of the (intended) device into (specifications of) simple(r) elements, the determination of physical properties of the device and the study of its functionality.

Synthesis - the building up of the simple elements found by analysis of an initial specification into an integrated whole intended to be a final specification.

The engineering process forms, speaking in IT-terms, a recursive process in which a specification of an object to be designed is transformed by analytic means into specifications of sub-problems - the parts of the object and their structural interrelationships - which are to be synthesized. The purpose of the whole process is to transform an initial functional description perhaps augmented with constraints pertaining to economic factors, time-limits and the like, into a full specification of an object that corresponds to the initial functional description, the way it is to be built and how well it meets other constraints and requirements.

The design process thus induces a natural hierarchy on the object to be designed. This process can be represented as a tree where specification, analysis and synthesis find their place in the nodes, representing a common level of abstraction, while the branches of the tree refer to the identified parts of the object. So, seen as a process, analysis moves control down the tree by generating subspecifications, while control is moved back up in the tree if the primitive and complex parts of the designer's product are considered acceptable with respect to the product's original specification. Control is then moved back up to the root of the tree by synthesis of the identified primitive and complex parts.

Ideally, a design process is started with only an initial specification of an object to be designed and eventually produced. This initial specification is analyzed so that subspecifications of identified parts are made. These subspecifications will then be analyzed (possibly by different specialists), in this way generating sub-sub-specifications until finally primitive parts are identified.

Although the idea of unrestrained top-down design may have its attractions, it is not realistic. Usually, "high-level" parts (e.g. engines) are available and easy to use. Given these complex parts it becomes attractive to restrain the analysis of a specification such that those parts are used when appropriate. This becomes especially advantageous when working in a multi-product environment: products are easier to make and maintain when they have more parts in common. Thus the design-tree becomes a design-*dag* when more than one product is considered. In synthesizing those common parts special attention will have to be paid to the multiple environments in which such a part is to function.

4.2. Specification.

The separation of the design process into specification, analysis and synthesis corresponds to the view explained by Simon (Simon, 1969) of artifacts as "interfaces" between an outer and inner environment. A specification describes the function an object should have ("a device to tell time"), perhaps augmented with constraints ("it should work at night", thus ruling out sun-dials.). The specification gives a description of the function of the object in an outer environment: it can be stated in terms of facts, demands, desiderata and constraints with respect to the environment in which the device is to operate. In other words, a specification gives a description of the functions that the specified device should have in an external environment and a description of the circumstances in which the specified device has to work.

The purpose of a specification is to present facts, demands, desiderata and constraints about the object to be designed in an orderly way to reduce opportunities for misunderstanding to a minimum. Although no formal specifications are implied here, it should be noted that informal specifications may easily lead to contradictions especially when they are more complex.

4.2.1. Structure of problem-specifications.

A specification, in the sense intended here, will be an informal one leaving much to decide to the engineer. Suppose that a company asked one of its engineers to design an inventory-control system with the constraint that none of the company's products will ever be out of stock. A literal interpretation of such a specification will probably lead to impractical solutions: infinite, or at least gigantic amounts of products would have to be stocked to meet the constraint.

The engineer will therefore reinterpret the specification to a more feasible version where an inventory-control system has to be designed with the constraint that the frequency any of the company's products are out of stock is reduced to an acceptable level, or minimized if possible. If the engineer cannot strive for minimization and has to settle for acceptability, he can, with or without help, decide what constitutes an acceptable level.

The engineer in the example has transformed an initial specification into a problem-specification: given the means the company has for storing products (buildings) and other means to extend its storing capacity (money), design a way to use these resources in an optimal, or at least acceptable way.

The engineer has actively participated in the formulation of the problem. This fact is typical for most hard problems engineers, architects, lawyers and other professionals face every day. Note that these problems are in marked contrast with the exercises we all had to solve in school: most school-problems in algebra, geometry, physics and chemistry leave little room for such manipulations: they are well-specified.

The difference between "well-structured" and "ill-structured" is - as it is with good and evil - much more a matter of degree than a matter of absolute dichotomy. Consequently, it becomes harder to separate the two. Simon (Simon, 1973) tried to describe well-structured problems in the context of artificial intelligence and cognitive psychology. He found that a problem is well-structured to the extent that it has some or all of the following characteristics:

1) There is a definite criterion for testing any proposal solution, and a mechanizable process for applying the criterion.
2) There is at least one problem space in which can be represented the initial problem state, the goal state and all other states that may be reached, *or considered,* in the course of attempting a solution of the problem.
3) Attainable state changes (legal moves) can be represented in a problem space, as transitions from given states to the states directly attainable from them. But considerable moves, whether legal or not, can also be represented - that is, all transitions from one considerable state to another.

4) Any knowledge that the problem solver can acquire about the problem can be represented in one or more problem spaces.

5) If the actual problem involves acting upon the external world, then the definition of state changes and of the effects upon the state of applying any operator reflect with accuracy in one or more problem spaces the laws (laws of nature) that govern the external world.

6) All of these conditions hold in the strong sense that the basic processes postulated require only practicable amounts of computation, and the information postulated is effectively available to the process - i.e. available with help of only practicable amounts of search.

According to Simon, these criteria are not entirely definite. Nevertheless, he checked them against several problem solving areas among which was the game of chess. The problem of the best single move can be regarded as a well-specified one. Functions can be devised that give scores depending on the quality of the move. As there are always several moves to choose from, the evaluation function can be applied to the alternatives to select the move with the best score. But matters are different when we consider a complete game from the first to the last move. First, we have to do with an opponent. Every move of the opponent adds information to the state of the game and also changes some expectations we have about the opponent. Second, an opponent may make unanticipated moves. Third, once a move is made, it is irreversible. Thus the problem of winning a game of chess, seen as a well-specified one, needs redefinition at each move, perhaps a slight one with anticipated moves, but all the more drastic with unanticipated ones. Therefore, although the problem of selecting a single, best move is well-specified, the problem of winning an entire game of chess is ill-specified.

Simon concludes: "In general, the problems presented to problem solvers by the world are best regarded as ill-structured problems. They become well-structured problems only in the process of being prepared for the problem solvers. It is not exaggerating much to say that there are no well-structured problems, only ill-structured problems that have been formalized for problem solvers. [...] Nevertheless, there is merit to the claim that much problem solving effort is directed at structuring problems, and only a fraction of it at solving problems once they are structured." (Simon, 1973)

4.2.2. Organizational context and problem-specification.

As it is, real-world problems are not very well-structured. It is all the more surprising that we appear to do well, even when designing very complex objects like airplanes, satellites, communication systems or information processing systems. The reason that we are able to build and improve such complex systems is that they are based on robust theories: it would be inconceivable that we launched a satellite and expected it to follow its designed course without reference to celestial mechanics.

Knowledge enables us to reduce significantly the "search space" in which the solution to our problem may be found. The solution of real-world problems usually involves a corporate effort. Real-world problem solving then becomes a distributed activity that needs coordination by an organizational structure. An organization, with its various departments in which areas of knowledge are bundled, adds by its very structure the information to an initial description by which a first decomposition into sub-descriptions can be made. The organizational structure itself takes (some) care of the structuring of an informal specification by allocating certain tasks to subgroups in the organization.

This leads to a more precise task-description for its members, making it possible to regard certain parts (made by others) as resources. The products and services a group is responsible for can be regarded in a more abstract way which often makes it easier to cast the problem in terms of abstract models.

The more independent the abstract models are, the less the number of ties and dependencies between the models exist. To the extent that various departmental tasks are independent, the departments can work in an independent, parallel fashion. However, a group is not completely at liberty to work as it pleases: there are always constraints to adhere to.

Constraints can limit the complexity of a task significantly: they "narrow down" the search space, so to speak. The engineer who transformed the inventory-control problem into an optimization problem a few pages ago could do so by including in his reasoning process specific knowledge about the organization that posed the problem. This information does not have to be directly relevant to the area of expertise of the engineer: facts of a financial, administrative or organizational nature can function as constraints to clear up an ill-specified problem. Constraints provide information that can serve to reduce the number of possibilities that have to be considered in solving a problem. Therefore, constraints can lead to better problem-specifications.

Constraints by themselves do not endanger the concurrency inherent in an organization. The least damage to concurrency occurs when all constraints are known before the actual work for the project starts. In that case all groups can work in a perfectly independent way and when every group works within schedule, the partial results can be synthesized at a moment planned in advance. Such a situation is ideal. However, it is unavoidable that groups produce information needed by other groups as the project develops: when a product's design is in an advanced stage reliable ideas can be formed about the consequences that a design must have for e.g. its manufacturing aspects and its price indication.

An organization cannot always decompose its work in a way that maximizes concurrency. The products of a department may not be as anticipated, thus influencing the work of other departments in an unforeseen way. And generally, the unplanned good or bad ideas of one group may, when accepted, heavily influence the work in other groups. So, flexibility of the departments in an organization appears to be mandatory. To be flexible, one needs to have information about new requirements and decisions as soon as they are decided upon.

4.2.3. Implications for a design-system.

As initial specifications are transformed by the design-process into final specifications that have all the detail needed, it appears that an engineering design system would have to be based on an information system that has to accommodate not only traditional database-items like parts, price and suppliers plus their relational structure, but it must also contain the informal specifications a design process starts with and documentation of decisions made at several stages of the design.

The information system will have to allow the "growth" of an informal specification into a final one. The transformation itself of the initial specification into the final specification will have to be representable in the information system so that all the participants remain informed about the exact status of the project(s) they are involved in. This includes the facts and constraints that have consequences for the work done emanating from other groups of the project. An information system for engineering design must represent the state of the project in a dynamic way: it also has the function of a news-medium.

Apart from having to represent the state of a project in a dynamic way, the information-system will have to accommodate the structure of the organization and the particular fine-structure the organization has chosen to carry out a certain project. Groups and departments will operate on and make changes to only a part of the whole information system, while other parts are to be regarded as data for inspection only and there are items in the system that have no concern at all to specific groups in a project. Hence the information system must be partitionable into subsystems for different groups. These partitions can then be tailored to the specific computational needs posed by their respective specialisms. The advantages are similar to the advantages present in the provision of database submodels. Date (Date, 1975) lists a few:

- simplification of the user's view,
- the same data may be viewed by different users in different ways,
- the submodels provide a way of preconditioning,
- data may be converted such that older applications are still usable,
- automatic security is provided for data of no concern to particular users.

The way in which these partitions are implemented can vary from a physical separation of the different aspects on different hardware either connected by network or not connected to a logical separation of these modules all working on a single machine.

4.3. Analysis.

A device can be regarded as an interface between an outer and an inner environment (Simon, 1969). The advantage of dividing outer from inner environment

> "... is that we can often predict behaviour from knowledge of the system's goals and its outer environment, with only minimal assumptions about the inner environment. An instant corollary is that we can often find quite different inner environments accomplishing identical or similar goals in identical or similar outer environments." (Simon, 1969)

Often there is a corresponding advantage in the division from the standpoint of the inner environment:

> "In very many cases, whether a particular system will achieve a particular goal or adaptation depends on only a few characteristics of the outer environment, *and not at all on the detail of that environment* [italics added]. It is an important property of most good designs, whether biological or artefactual. In one way or the other, the designer insulates the inner system from the environment, so that an invariant relation is maintained between inner system and goal, independent of variations over a wide range in most parameters that characterize the outer environment." (Simon, 1969)

A specification - whatever its level of formality - can be stated in terms of the functions the device should have in the outer environment. Analysis is concerned with the design of an inner environment: an anatomy of the proposed device is produced. This anatomy contains a structural specification of the device and based on this structure (understood as an external environment) a functional specification of the identified parts can be given.

A functional specification shows *what* is needed in terms of an outer environment; a structural specification points out *how* the qualities mentioned in the functional specification can be attained in terms of an inner environment. A structural specification contains two distinct components: one component lists what parts are needed in a design and the other component specifies how these parts are related: it describes the structure, the skeleton of the device.

4.3.1. Structuring.

The structural part of a design describes, abstracted from the parts it has, how the required characteristics of the device are obtained. It describes the inner environment and from this description the correct or incorrect functioning of the device can be inferred. At the same time it describes an outer environment in which the identified parts are to function. In this sense an artifact is truly an interface between an inner and an outer environment. Note as well, that the identified parts are a level lower in abstraction than the device being designed at the current level: it is in this way that the design-tree is traversed. The designer, working in a node of the tree, can concentrate on the logical and structural parts of his design and can regard the parts simply in terms of the function they have in his device, whether these parts are themselves simple or complex.

In relation to our phenomenon of structure/parts dichotomy, Pratt (Pratt, 1977) considers the Competence/Performance dichotomy in programming and Kowalski (Kowalski, 1979) coins his equation "Algorithm = Logic + Control".

As a general rule one could say that a design is better to the extent that it has less structural complexity with the same functionality, or, conversely, that a design is better to the extent that it has more functionality with the same structural complexity.

Improving a design, thus simplifying its structure while leaving its functionality invariant or enhancing its functionality under invariance of its structural complexity can be done by special knowledge about the characteristics of the parts and integrating their functionality into the structure of the design. This is an important aspect of engineering. Improving a hierarchically decomposed structure of a design can invalidate the hierarchy, as these improvements are based on using special knowledge about parts of the design. To concentrate on the logical structure of a device, a designer abstracts from the parts, viewing them as immutable black boxes. When the logical structure has emerged, a good designer can integrate the structure of the parts into the structure of the device itself in order to improve the whole design. The strict hierarchy of the design may no longer exist after these modifications. Just as Simon (Simon, 1969) speaks about near-decomposability of systems found in the real world, Sussman and Steele (Sussman, 1980) speak about almost hierarchical systems in engineering design.

An example is the design of a clock. A clock can be functionally decomposed into an energy-source, an oscillator that emits pulses at regular intervals and a scaling device to transform the intervals to desired units; these parts being connected with transmitters. If it is decided to build a mechanical clock, then a spring could be used to represent the energy-source. A chain could transport the energy to the oscillator; the oscillator could be represented as a balance. A wheel-train could be used to scale the pulses from the balance and to drive the hands of the clock. This design can be improved by noticing that the chain to transmit energy from the spring to the balance and the wheel-chain can be unified into one wheel-chain, as a wheel-chain is capable of both transmitting energy and scaling, thus simplifying the design and making it possible to reduce the physical dimensions of the clock and perhaps making wrist-watches easier to wear.

4.3.2. Modelling.

When a design problem is extracted from a specification engineers can have much knowledge about the problem area or the problem can be unfamiliar. When a problem is familiar an engineer habitually uses a set of candidate structural models. These structural models are general (one could call them topologies for a device) and thus have many undetermined parameters. When a problem is less familiar, it must either be reformulated into a familiar one or be decomposed into a combination of problems that are more familiar. The problem here is, as always, to find a decomposition such that the parts are related in a well-controlled manner.

Whatever the familiarity of an engineer with a problem, the key to a design is the structural model of a device. Hence it is natural to base an engineering analysis system on the abstract models used in a particular engineering domain. It will not only have the *representations* of the designed device available in terms of the abstract concepts the user is familiar with, but also the *communication* between the designer and his (domain-oriented) design-assistant can be handled in terms of these domain-specific models. At the same time, a user does not have to learn to translate his domain-oriented concepts into computer-oriented concepts and vice-versa which is one of the major bottlenecks in the use of computers in application domains.

An example where representations are chosen to reflect the abstract models used is given by the work done in the Production Automation Project at the University of Rochester (USA) (Requicha, 1980). The example applies to the whole area of solid modelling. The model is based on the observation that very few subsets of three-dimensional Euclidean space are adequate models of physical solids. Suitable models for solids are subsets of three-dimensional Euclidean space that are bounded, closed, regular and semi-analytic; they call these sets *regularized sets*. Regularized sets may be viewed intuitively as curved polyhedra that can exist in the physical world around us. Regularized operations on regularized sets are defined as well to make more complex designs with guaranteed counterparts in the real world. In this way users are prevented from constructing devices that are impossible to realize.

An example where the communication between user and system is done in terms of domain-oriented models is reported by Barstow et al. (Barstow, 1982), who discuss a system to assist in testing qualitative log interpretation models against log data from oil wells. The user of this system describes a model by interacting with it in the natural terms of the domain. The system then writes the necessary software to implement the model the user wanted. They report that the entire process takes from ten to twenty minutes and achieves results which previously required two to three weeks of effort. An argument is further given against an application oriented user having to encode his abstractions in the formalisms developed by and for computer scientists under the motto "No matter how high the language, it's still programming".

Abstract model representations and user-system communication in terms of abstract models will evidently facilitate the use of computers as scribbling devices, where engineers can perform computations and try out small variations of a design more easily and with more speed and precision than without a design-assistant. But being based on abstract models, the system can handle questions having to do with less superficial aspects of a design. As an example the "what-if" type of dialogue can be mentioned, where far-reaching consequences of design-alternatives can be brought into view with relative ease on the part of the user.

The use of specialized knowledge does not only facilitate communications with a system, it can also make a system take over certain tasks or make certain tasks computationally feasible. Among the early examples the work of Waltz (Waltz, 1975) must be counted who describes a computer vision system that "understands" line drawings of scenes with shadows. By carefully enumerating the combinatorially possible labelings of lines and junctions and the physically possible labelings Waltz arrived at constraints on the combinatorics so enormous that it became feasible for a program to analyze these data and "understand" them in terms of simple solids.

Other highly successful programs were built by using specialized domain-specific algorithmic knowledge, heuristic knowledge and constraints. Examples of such systems are DENDRAL, a chemist's assistant, MACSYMA, a scientist's and engineer's assistant, MYCIN, a medical doctor's assistant and PROSPECTOR, a geologist's assistant. It should be mentioned that all these programs, though set up as research efforts, are now in productive day-to-day use. According to Traub (Traub, 1979) by 1978, some 25 papers have been published in major journals of chemistry, reporting results and the knowledge that had to be given to DENDRAL to obtain these results.

So, in order to build a system that is most useful as an engineer's assistant, the *abstract models of the engineering domain* must be used

- in the representation of the objects of interest,
- in the communication between user and system,
- to build a system capable to obtain more accuracy than humans normally do.

It must be stressed that the use of computational models, which results in programming must be discouraged by such a system.

Note that the domain-dependence of the designer's assistant coincides well with the partitions of the information system mentioned in section 2.

4.4. Synthesis.

Having developed a structural design of a device, the problem remains to *verify* whether the proposed design meets its specifications. The design, describing an inner world in structural terms, needs to be checked against the functional specification stated in terms referring to an outer environment. A complicating factor is that the functional and structural decomposition will usually not be the same, as exemplified by the clock in section 3.1.

Accepting or rejecting a plan with respect to specifications is not a simple matter. It is not realistic to expect that this task can or will be done entirely by machine. The reasons for this are not only that the specifications used in many areas are not formal enough, but that decisions are to be made by humans. An example of an area with knowledge formal enough that tests can be developed such that accepting or rejecting of a model depends on the value of a single parameter is statistics. Test statistics, as well as the probabilities that they exceed a certain value can all be computed easily. But the decision to accept or reject a model as explaining certain data is made by the statistician, and not by the computer. By their nature, decisions are hand-made.

Checking whether a design meets its specifications can only be done well when the specifications are formally stated. When informally stated checking a design can easily become impractical as it implies searching for a single case where the designed system breaks down. Informal specifications make it in general very hard to infer the conditions where a system must collapse.

To the extent that structural designs are more standard, their functional characteristics are better known so that most of the synthetic work will concentrate on constraint-satisfaction. These activities can be undertaken by using mathematical and physical models. A major limitation is the accuracy with which parts of an engineering discipline can be modeled.

An example of an area with incomplete knowledge is civil engineering. The properties of the important materials such as soil, sand and concrete are only partly understood in the sense that we cannot simply provide a set of equations and say that by the model exactly such and such behaviour can be expected. Using models in these areas often means making significant and involuntary simplifications.

Mechanical engineering is an area where much more is known about the materials, although important processes such as wear, vibration and metal-fatigue are not (yet) adequately understood.

The area of electrical engineering provides many very detailed models. But detailed models have their own difficulties: they may lead to computations with a complexity that is well beyond practical bounds. So electrical engineers are often forced to simplify their models in the hope that the outcomes remain accurate enough for the task at hand.

Apart from modelling through equations, which is well applicable when we know *what* a system *does* (external environment), simulation appears to be useful in studying the behaviour of complex systems where we know more about *how* a system *works* (internal environment). Parts of systems may be understood well enough to be modelled, but their combinations may easily become very complicated, as exemplified by devices developed in electrical engineering or programming. Besides modelling or using approximations to models, simulation may then be an applicable technique. As we often know how different components are combined in a device, we might as well use this information augmented by the models of their parts and make simulations of the complex device.

Synthesis can be subdivided into four general areas: structural synthesis, static synthesis, dynamic synthesis and the study of the consequences that the synthesized object has for its manufacture. Note that by means of these four types of synthesis the design itself must be synthesized with the total information system. The synthesis of the data from a partition with the total information system is the purpose of the synthetic phase, as seen from an IT-perspective.

Structural synthesis.

The question whether a device responds to its specification without special reference to the environment in which it is to function is a difficult one as has been argued in the previous paragraphs. This type of synthesis is concerned with the correctness of the structural design of a device and one of the main tools in this area is simulation.

Simulation in many cases needs to be further supported by diagnostic tools for interpreting the results. For instance, when a model does not perform as expected, most simulation methods fail to identify the source of the error. Diagnostic tools are application-dependent. Especially for interactive systems with simulation capabilities debugging facilities such as tracing and stepping appear to be important design aids, as users can stop the execution of the simulation as soon as things start to go wrong or inspect the state of the simulated system at every step.

Static synthesis.

During analysis an object is decomposed into physically distinct parts. Evidently, these parts have to be built up to synthesize the device. Static synthesis provides information to verify that a part does or does not satisfy its specification. To check whether all the parts actually fit together is important before the object is physically built. Systems have already been designed to detect intersections between objects. Boyse (Boyse, 1979) reports the design and implementation of such a system. The traditional two-dimensional drawings however will not always show intersections among three-dimensional objects, so that a three-dimensional graphics system must be part of a static synthesis system.

Another important function a static synthesis system must perform is the computation of static properties of objects such as weight, volume, moments of inertia, &c.

Dynamic synthesis.

Many engineering objects have moving parts. It is therefore, just as in the static case, important that the unwanted interference among these objects is detected before the object is physically realized. The system discussed by Boyse (Boyse, 1979) has also provisions for collision detection. It is evident that dynamic synthesis also provides information about the satisfaction of the specification of a part.

Consequences for manufacturing.

Synthesis is concerned with building up of elements to form a complex whole. Building a device does not have to be done to check whether all the parts fit together, but when the sequence in which the elements are put together is stressed, an assembly sequence is obtained. A synthesis system should assist in designing this sequence.

As a side-effect of these operations the generation of robot-programs can be envisaged. The information generated by synthesizing an object plays a role that is comparable to the role of the constraints in the analytic phase of the design.

4.5. An example.

An integrated engineering design assistant should be capable to perform tasks such as the automatic generation of assembly commands for industrial robots.

Suppose an engineer has designed a mechanical object including its parts and their structural relationships. The parts are represented in terms of solid models with their own coordinate systems. Similar data are present to describe the structural relations these parts are to have. The engineer can, when synthesizing the object, specify an assembly sequence from which the exact positions of several parts at various stages of the assembly can be inferred and checked for consistency. When this is done, control commands for an industrial robot can be generated; this should lead to a much more efficient and less error-prone robot programming process than the techniques currently in use.

Apart from possible error-correction - perhaps due to inconsistencies in assembly - the only human intervention in this example is the engineer specifying an assembly sequence. As some assemblies are complex, the help of a computer would not be unwelcome or impossible here. It could aid in establishing an optimal assembly sequence.

4.6. Towards design rules for a design assistant.

In the previous sections the design process has been subdivided into three activities: specification, analysis and synthesis. We believe that this paradigm can lead to an implementable design system.

4.6.1. Specification.

From the discussion of the activities related to specification it became clear that in order to understand and carry out a complex design task we need organizational support in the form of specialized departments and perhaps other, less permanent organizational structure. From an IT-perspective this would lead to an information-system that allows partitions according to coarse and finer structural characteristics of an organization or project.

This information-system will initially contain many informalities (the objects and parts not yet fully designed), perhaps stated in the form of natural or structured language. It will also contain representations of devices that can be produced, are in stock, etc. and the more usual database items such as prices, part-numbers and suppliers including their relational structure.

It will also have to accommodate the informal specifications, either stated in natural or structured language. Then, there must be provisions to document decisions made during design.

Another very important characteristic of such an information system is that it reflects the state of the project *in a dynamic way*. Only in this way can such a system be used to its full potential.

Partitioning this information-system can be done in various ways. It can be done in a virtual way, such that the system appears to manifest itself in separate sub-systems but is hosted in a single mainframe computer; it can also be designed in a physically distributed way where each subtask and corresponding partition of the information system is implemented on a different machine, with or without a network-connection. Whatever the realization of the system, it will have to be able to synthesize the work coming from different partitions, especially checks for inconsistencies appear to be desirable in the presence of (initially) informal specifications.

4.6.2. Analysis.

As to the analytic phase a designer's assistant would only be acceptable if representation of and conversation about the object being developed would be in terms of the abstract models used in the engineering domain of the user (i.e. the designer). Although all computational tasks will ultimately be mapped into a finite combinatorial structure (a computing device), this combinatorial model is far too general to be of practical use for specific applications by computer non-professionals. It would be better if the design system would present itself in terms of concepts and structure familiar to the designer. A designer working in such an environment can be expected to have less trouble learning to work with the system and would hopefully have more pleasure using it. This design rule would also prevent a designer to have much opportunity for making more errors than are usually made using the abstractions of an application domain: the problems associated with application programming are avoided.

4.6.3. Synthesis.

In the synthetic phase the abstract domain-dependent models of an engineering field are just as important as in the analytic phase. The detection of errors in parts of a design and the detection of structural errors in the design itself are best communicated in terms of known concepts that users are well familiar with.

A synthetic system can be used for various purposes. It can be used to check whether the parts do fit together in the ways expected and to compute various static properties of the device (static synthesis). It can be used for the detection of collisions among moving parts (dynamic synthesis). Another important function a synthesis system should be capable of is the overall check whether a design conforms to its specification (structural synthesis) - the search for structural errors is certainly a synthetic activity: if the design proves to be unsatisfactory, it may be followed immediately by a new problem analysis. A fourth area is the study of the consequences a design has for its manufacture. Evidently, just as an analysis system a synthesis system must represent its data and communicate with its user in terms of the models of the relevant engineering domain. it must have good diagnostic and debugging capabilities in terms of those models.

Although we have identified these four areas they may not all be relevant in every engineering activity. As the difficulties to be expected for complete structural synthesis can be substantial, static and dynamic synthesis taking place higher up in the design-tree can play an essential role in the detection of structural design errors in the parts causing trouble.

The four types of synthesis together must take care of the integration of the data representing the design into the overall information-system. Besides static, dynamic and structural checks, a synthesis system will have an important function in maintaining the overall consistency of the object the synthesized design is to be part of.

4.7. A processing strategy for an engineering design assistant.

Having separated the design process into the aspects of specification, analysis and synthesis, it is natural to treat the consequences for the processing strategy along the same lines. What is needed is assistance in a difficult, costly and error-prone process.

Engineering design is a very broad multidisciplinary area. The treatment of a processing strategy can therefore not be too specific, as there are so many different domains all having their own special models and methods of work. Moreover, any specific design project itself is multidisciplinary: it not only contains topics typical for an engineering discipline, but also other factors, administrative or otherwise.

However, one of the most fundamental requirements for a design assistant for any specialism is that it must be implemented as an interactive system. Clearly, the rate of alternation between specification, analysis and synthesis can be very high; it could even be stated that they are hardy separable in some circumstances. It is not difficult to imagine an engineer inspecting the external environment of the device to be designed, while the device itself is being developed by doing simulations to get an initial idea of how such a device might be constructed (rapid prototyping). To be of a valuable assistant a design-system must not only present itself in an interactive fashion for tasks that are relatively fixed, but it must also allow rapid context-switching.

4.7.1. Specification.

The transcription of the specification seen as part of a design may be directed towards many aspects or environments, such as the description of the external environment of a device, the materials and parts to be used, the manufacturing methods available to a factory, assembly constraints, the interface to other parts and even aesthetics. A design assistant therefore requires great freedom for the designer to switch contexts during the design, because all the aspects of a device have to be taken into consideration.

It would be naive indeed to expect that a design specification as seen and generated from one aspect would automatically generate the properties and specifications as seen from other aspects. A designer will, during analysis, verify a certain aspect (perhaps by switching context often) and the results of this may influence further design decisions in the global context.

The tools for combining specifications from various sources into complete specifications for a given aspect as well as a canonical (domain-dependent) representation require the most advanced specification techniques currently available. And they must be further enhanced.

The techniques and tools for this stage of the design characterizes design support as a highly interactive subsystem for handling complex data. The architecture of a design system for CIM should reflect the three key activities. The interactive dialogue systems and the information system support should allow easy context switching. Such a system will require dedicated processors with strong central backup for the data needed when switching context. Needless to say that much processing that is typical for database-systems is required here.

The implementation of a system as sketched here may appear to be a very ambitious goal, but it merely requires a data- and processing organization that needs to be present for the integration purpose of CIM anyway. On the other hand, realizing such a goal would exhibit the integration necessary for its users and is therefore considered to be indispensable.

4.7.2. Analysis.

The analytic phase of the design process is concerned with adding levels of detail to a given specification until finally, on reaching the leaves of the design-tree, the primitive parts of a device are identified. For each subcomponent this level may be chosen to be different (e.g. a designer may decide to use existing components rather than designing new ones.). In the analytic phase therefore, access to a partition of the information system where components and parts can be inspected is vital. Next, the information about the relational structure between components must be created and manipulated. A third essential aspect is access to a partition of the information system where data about manufacture and assembly can be communicated, so that the engineer not only designs a device, but designs a device that can be manufactured and about which cost calculations can be done in an early stage.

Hence, analysis is primarily concerned with interactive construction of parts and components during which many database transactions can take place.

A most important aspect of an analysis system is the construction of the representations of the objects that play a role in the design. Design, whether technical or artistic, is most naturally done with graphical means. The dialogue modules of a design system will therefore have to be centered around a graphic system. Especially important for analytic design work is the aspect that is concerned with input of the objects to be manipulated. As stressed before, this is most easily done with the symbols developed in different domains and specialisms. This means that a designer's assistant must be capable of "understanding" these symbols. These systems will therefore have to be equipped with graphical languages to enhance the communication between user and system. These languages must guarantee that the objects described by the user are transformed into the canonical, domain-dependent representations that are deemed necessary for a successful cooperation between user and system.

Another important tool in such a system would ensure that to each component the associated information relevant for other contexts is maintained, asked for or added to as needed, work that is often very demanding, boring and error-prone. Another important module could maintain the consistency of the overall model during refinement as far as possible.

Apart from the advanced input facilities, analysis puts demands to processing similar to those of specification.

4.7.3. Synthesis.

As the four types of synthesis together must accomplish the synthesis of partitions taken from the information system back into it, database support is a major general aspect of the processing strategy for synthesis.

Structural synthesis.

In the synthetic phase simulations is the most important tool to verify the structural aspects of a design. The data for a simulation run are to be derived automatically or derived from the designer's indications from the synthesis of the refined structural components. Current state of the art is such that simulation is the *only* tool for practical verification.

In future systems formal specifications on several levels might either obviate simulation in the areas formalized because the checks can be calculated or at least the data for the simulation can be generated automatically. In both cases (checking and simulation) the processing typically involves much number-crunching, perhaps enhanced by graphical output for visual verification.

Simulation must be enhanced with diagnostic aids to find out *why* a simulated system breaks down. The reasons for collapse are not always immediately clear. Building simulation system with diagnostic capabilities can only be a comprehensible task when the organization of the data on which these systems must build is very well structured.

Static and dynamic synthesis.

Checking whether the parts of an object can be put together either statically or dynamically implies a good three-dimensional graphics environment, so that the designer can see intuitively where things go wrong. The type of computation is such that again a dedicated graphical workstation with good capabilities for three-dimensional interaction and numeric processing is needed to perform these functions in a satisfactory way.

Consequences for manufacturing.

The data generated with the design of assembly sequences and other manufacturing aspects directly dependent on the designed object generates a partition of the global information system. This information has to be integrated with the total information system so that it can be integrated with other factory-dependent data that will form a new partition. Good database facilities therefore appear to be essential here as well.

References

Barstow, 1982. Barstow, D., Duffey, R., Smoliar, S., and Vestal, S., "An automatic programming system to support an experimental science.," pp. pp 360-366 in *Proceedings 6th international conference on software engineering, Tokyo, Japan*, IEEE Computer Society, Los Angeles, California, USA (September 1982).

Boyse, 1979. Boyse, J.W., "Interference detection among solids and surfaces.," *Communications of the ACM* **22**(1) pp. 3-9 (January 1979).

Kowalski, 1979. Kowalski, R., *Logic for problem solving,* North-Holland Publ. Comp., Amsterdam (1979).

Requicha, 1980. Requicha, A.A.G., "Representations for rigid solids: theory, methods and systems," *Computing Surveys* **12**(7) pp. 437-464 (December 1980).

Simon, 1969. Simon, H.A., *The sciences of the artificial,* The M.I.T. Press, Cambridge, Mass., U.S.A. (1969).

Simon, 1973. Simon, H.A., "The structure of ill-structured problems," *Artificial Intelligence* **4** pp. 181-201 (1973).

Traub, 1979. Traub, J.F., "The influence of algorithms and heuristics," Carnegie-Mellon University rep. CMU-CS-79-102 (1979).

Waltz, 1975. Waltz, D., "Understanding line drawings of scenes with shadows," pp. 19-93 in *The psychology of computer vision*, ed. P.H. Winston, McGraw-Hill Book Company, New York (1975).

Sussman, 1980. Sussman, G.J. and Steele, G.L., "CONSTRAINTS - A language for expressing almost hierarchical descriptions," *Artificial Intelligence* **14**(1) pp. 1-39 (1980).

Pratt, 1977. Pratt, V.R., "The Competence/Performance dichotomy in programming," pp. 194-200 in *Proc 4th ACM Sigact/Sigplan symp. on Principles of Programming Languages, Santa Monica, Calif.*, ACM (January 1977).

Date, 1975. Date, C.J., *An introduction to database systems,* Addison-Wesley Publishing Company, Inc., Reading, Massachusetts (1975).

5. Flexible Manufacturing Systems

Since the industrial revolution there have been two basically different ways to organize a production process. The first is mass production where a single product can be produced in large quantities for relatively low costs. In a mass production organization it is not possible, or only at great expenses, to change the product. When the quantities that could be sold were not large enough to justify the setup costs of a mass production process one had to resort to manual production, the second way to organize a production process. This manual production has been supported with increasingly powerful and precise machinery. The human task consisted then of operating and controlling this machinery. The current revolution in information technologies has made it possible to transfer these functions from humans to computer systems. This is the first step toward flexible manufacturing without human intervention. The second step is to organize the production process in such a way that optimal advantage can be had from the flexibility available in the computer control. The motivation to set up a flexible manufacturing system is to combine the advantages of a fully mechanised (or, in modern terms, fully automated) production line, which are:

- Little dependence on experience of workers;
- Large flexibility in production rates possible without employment problems;

with those of manual production:

- Ease of switching to a modified version of the product;

The term "FMS" is rather loosely used in industries. This is because many types of flexible manufacturing systems are in practical use with various automation levels. This is reflected in the use of the term FMS in literature where a variety of definitions can be found. These definitions differ in the accents which are laid on the amount of computer control, the automation of material transfer and handling, and the amount of flexibility achieved. We will base our definition on the one given by (Ohmi, 1982) in their survey study of FMS in Japan. Their definition restricts FMS to machining operations and although this may be the most important application of FMS today, FMS techniques can also be used for operations like spray-painting, welding, and assembly and therefore we have reformulated their definition in such a way that it is no longer restricted to machining.

The term **Flexible Manufacturing System (FMS)** means the production system mainly consisting of a computer controlled machine with the considerably automated equipment for transfer/handling of workpiece/tool, under control of a computer or corresponding device, and being adapted to the flexible and adaptive parts production.

Ohmi e.a. also give the following subdivision.

FMC (Flexible Manufacturing Cell) : the FMC consists of a NC machine tool with a pallet pool or a magazine. In general this type can be operated without any workers during one shift or several hours.

FMC shop : the FMC shop comprises plural FMC's, where the transfer of workpiece is performed by workers between them.

Pure FMS : the typical system where NC machine tools are linked with the automatic material transfer/handling means, which assures the flexible and automatic loading of the workpiece.

FTL (Flexible Transfer Line) : a number of NC or head-changeable machine tools are combined by the automatic material transfer/handling means. The FTL produces a family of parts without the flexible routing of workpieces.

FMS creates an alternative for inflexible automation on one hand and manual production on the other hand. The benefits of FMS are different when compared with each of these extremes. When the alternative is manual production then the result of the introduction of FMS is a decrease in flexibility and an increase in efficiency and productivity, whereas when the alternative is inflexible automation the advantage of FMS is increased flexibility and an increased economical life time of the production system. Compared with manual production the advantages can be split up:

- Reduced work in progress, due to the lack of need for small stock in between operations to account for changes in the speed humans work with.
- Reduced stock levels
- Faster throughput times. It turns out that the automation of material transfer and handling decreases the throughput time.

The increase of flexibility compared with inflexible automation is obvious. The increased economical life time of the machinery is due to the increase in flexibility. Because of that the machinery can remain in operation after a change of product design whereas with inflexible automation the machinery can be thrown away after a change in the product design. Especially in those cases where the quantities of production are barely high enough to justify inflexible automation chances are large that at the end of the economical life time of the machinery they are still technically in perfect order.

5.1. The design of Flexible Manufacturing Systems

When designing a flexible production line the first key question is for which range of products it is intended. This determines the amount of flexibility needed. The problem of the designer is that when he makes the production line more flexible then needed he might waste money, and if he doesn't make it flexible enough it might not be capable to make the products that the buyers want. Early FMS systems implemented at the second half of the sixties in the USA offered a large flexibility (up to over 100 different parts) but it turned out that the expected productivity could not be reached. In the years thereafter only systems with a flexibility between four and ten different parts were installed (Spur, 1982). To come to a correct analysis about the range of products which need to be produced information about future trends in the market, technological developments, and about the capabilities of alternative FMS schemes has to be carefully evaluated. When this is not done with sufficient care serious problems may arise after installation of the system. Bryce reports in a study on FMS in the USA that

> "Most companies thought they had bought more flexibility then they really had. In the worst case design changes to the components to be machined meant that only 3 of the originally intended 10 could be processed on the system by the time it was delivered. While this demonstrates poor communication it also indicates a serious lack of proper task definition by customer and vendor at the conceptual stage." (Bryce, 1982)

The task of the product designer consequently changes since it is no longer sufficient for him to design a single product but in cooperation with the designers of the production process he has to design a range of products. If it is known in advance what the products will look like, it is possible to determine whether on a particular flexible production line these products can be made. In practice however it is not possible to determine precisely how the products that have to be made will look like. A particular flexible production line defines a range of products that can be made on that production line. Since it is not possible to determine exactly which products one will want to make it is also necessary to try to find out whether those products wich are not within the range of a proposed FMS line don't have to be made either.

Choosing machinery for a flexible manufacturing system will be influenced not only by the direct requirements of the production line but also by the perceived life time of the production line. When it can be foreseen that the lifetime of the machinery is longer then the time the production line is operational it is important to what extent the machinery used on that production line is typical for that production line or whether it can also be used for different purposes. When, for example, for a specific operation in the production line a robot with four degrees of freedom would be sufficient, but that to have a reasonable chance that the robot could be used elsewhere it would need eight degrees of freedom, then it might be more economical to use the one with eight degrees of freedom.

A flexible manufacturing system is only feasible when the flexibility is achieved with a small and limited number of tools. This requires a large degree of standardization in the production process and this will be reflected in the design of the product. This interaction between the mode of production and the design process is of great importance for the design of computer integrated manufacturing systems. In general the product designer has to get accustomed to the limited possibilities an FMS offers compared with manual operation. The usage of computer aided design and product engineering can be great help for the designer since they can give him direct feedback on the feasibility of his design. The obvious way to achieve this is to have a model of the FMS incorporated in the CAD system. For the production engineers such a model needs to be more refined than for the designers (although, of course, it doesn't hurt to have an extensive model available for the designers as well). The production engineers need the model to simulate the production and for them the model also has to include a possibility to have the model simulate testprograms for the computer controlled machinery.

5.2. Numerical Controlled Machines

In machining there has been an trend towards "machining centers" capable of doing all sorts of machining, like drilling, milling, grinding, etc. To switch from one operation to an other they are equipped with tool magazines. The result of this development is that on one such a machining center all operations to machine a product can be done. The flexible automation of machining systems is largely based on the NC - technology. What needs to be added to achieve automated production is workpiece-, tool-, gauging- and clamping handling, which are done manually with traditional NC machinery.

The programming of NC machines is mostly done with the programming language APT. APT is language which even predates Fortran and it has a syntax which is today hardly acceptable.

5.3. Automated transportation and loading.

Very often the workpieces aren't transported individually but in groups, fixed on a pallet. The reason for transferring the workpiece fixed on a pallet is to facilitate the positioning of the workpiece when it arrives at the next machining center (apart from, of course, the desire to diminish the number of transport operations by transporting a number of workpieces at once). When the machining centers are sufficiently close no separate transportation system is needed and robots can transport the workpiece from one machine to an other directly or through a buffer area. When a separate transportation system is necessary robots are usually used to load the workpiece from a transportation system on the machining center and vice versa. Also special purpose devices are used.

The transportation systems can be classified according to the constraints which they place on the route of the workpiece. An automatic carrier without rails (wire guided) imposes a minimum of constraints. The system layout determines the routes a carrier can follow. This system layout can be changed easily. Within the constraints of the system layout the carrier is free to move along any route. This route can be decided upon by a control system during operation of the FMS to optimise machine usage and transport throughput. Consequently this type of transportation system is becoming increasingly popular. Next to highly flexible transport systems there exist also inflexible ones such as a pallet carousel or a conveyor belt. The disadvantage of such systems is that they fix the route of the workpiece and in many cases they also pose constraints on the cycle times of the machinery operating on the workpiece.

5.3.1. Robots

Robots are used for spray painting, spot and arc welding, and loading, unloading and clamping. Sometimes they are also used for assembly, but this application is rare. Most robots are used currently without sensor systems and fall in the category "dumb robots". Consequently a great deal of effort has to be made in correctly positioning the objects on which the robots operate on, since the robots are not capable to adapt their actions when the object is (slightly) misplaced.

Generally robots consist of an "arm" with several joints and on the end of the arm an "end-effector". This end-effector might have hand like properties but in a lot of applications (welding, painting) this end-effector has no resemblance with a human hand. The most popular drive systems for robots are electrical engines and hydraulical systems. The hydraulical systems are used when heavier objects have to be moved.

5.3.2. Robot sensor systems

Usually robot sensor systems are not typical for robot applications but can also be used as stand alone systems or in combination with other computer controlled machinery. This is especially true for vision sensors. Therefore sensor systems are treated separately. At this point we will only stipulate some important and typical usage of sensor systems in robotics. A complete survey of sensor systems can be found on the chapter on that topic.

Force sensors are used to determine the forces which are being exerted on the robot. Research interest on force sensors focuses on the development of force feedback systems where the output of the force sensor is used to adapt the actions of the robot. This is mostly applied to achieve a so-called compliant motion, where the robot moves under the constraint that a force is continuously exerted. To realize a compliant motion by means of force feedback puts a heavy demand on the available real-time processing power. It is necessary that roughly every millisecond the output of the force sensor is evaluated. This is hard to realize with current microprocessors when they are simultaneously used for other processing needs. In industrial applications no attempts have been made to realize compliant motion by means of force sensors.

As part of a robot's hand tactile sensors can be used to acquire information about the object being held. For this purpose matrices of small (a few millimeter cross section)touch sensors have been developed in research institutes. So far no industrial use has been made of such systems.

5.3.3. Programming Robots

Robots can be programmed in several ways. The main categories are Guiding systems, robot programming languages and planning systems (Lozano-Perez, 1982).

Guiding

In guiding systems a sequence of positions is recorded which can be "played back". Guiding systems differ in the way robot positions are specified and the possible motions between positions. The most common way of specifying positions is by moving the robot to the desired position by hand, either directly or via a master slave linkage. Motions can either be specified by moving the robot through the motion directly or by choosing a predefined way of motion between points (e.g. a straight line or uniform motion of the joints). Some systems offer the possibility of a guarded motion. During a guarded motion the robot moves until a sensor condition becomes true. Additionally some guiding systems offer simple control structures(e.g. the commercial ASEA robot). These enable the programmer to define parts of the guiding sequence as subroutines. For example it is the possible to define the picking up of an object after the robots hand has been positioned above it as a subroutine. When several objects have to be picked up from a pallet one can move the robots hand above one of these objects and then call the subroutine to pick it up. The advantage is that the programminging takes a shorter amount of time. A functional advantage can be achieved when it is possible to check external signals and decide later which part of the taught sequence should be executed. This requires the availability of conditional branching next to defining subroutines. When a CAD system is available which contains a model of the robot and the environment it is possible to simulate the programming operations on that system (off-line guiding). Such a simulation system can also be used in combination with a robot programming language.

Programming Languages for Robots

A robot programming language is basically an ordinary programming language (like Fortran, Basic or Pascal) with extensions to make it suitable for the programming of robots. Several of such languages have been developed by research institutes, robot designers (IBM, Unimation) and large users (Mc Donnel Douglas). The languages differ in their sophistication as an ordinary programming language as well as in their extensions. The extensions commonly available are commandos (usually in the form of predefined subroutines) for a move to a point in Cartesian space, a guarded motion and for compliant motion; furthermore there are facilities to obtain sensor information and to interrupt an action (e.g. a motion) due to sensor information.

Task Planning Systems

A programmer using either a robot programming language or a guiding system has to specify the motions and actions which the robot should carry out. The goal of task planners is to determine the necessary robot actions on the basis of a description of the task to be performed. The output of a task planner is a program similar to one which a programmer has to make himself when he uses a robot programming language. Task planners constitute therefore a higher level then both guiding systems and robot programming languages. Task planners require a geometric and physical description of all objects in their environment. If a CAD system is used for the design of the product a large part of this description might already be available.

Discussion

Guiding (play back, teach in) is by far the most commonly used in commercial applications. It's main advantages are that no programming experience is required and that the result is immediately visible. It's main drawbacks are that programming cannot be done before installation of the robot and that it is difficult to handle sensor information. Robot programming languages are not used in commercial applications today but will become important because of their capability to handle complex sensor information. The disadvantage is that programming experience is required. The importance of planning systems is that they may provide the link to automatic generation of robot programs. Currently there are no planning systems which can be used commercially. This is partly due to the high level of integration of design and production necessary to have the description of the environment easily available to the planning system. The other reason is that there are many difficulties in constructing a planning system (in fact, no complete system has yet been implemented).

Currently there is a clear trend to extend the possibilities of the guiding systems by embedding the guiding in a simple programming language. Clearly, the more possibilities these simple languages offer, the more they will start to resemble the robot programming languages used in research institutes. For simulated guiding the situation is similar. It needs to be possible to tell the simulator that the robot should go somewhere, one wants to define subroutines, include conditional branching on sensor information, etc. This means that the input language for the simulator will have to offer the same type of facilities as a robot programming language.

The main processing need of a robot is the real-time interaction with the servo control. Secondly it is necessary that the robot is capable to recalculate the trajectory when the sensor system indicates that the object to operate on is slightly misplaced. Currently robots require expensive positioning systems to ensure that the object is precisely at the right place. It would be a considerable improvement if a sensor system could be used in combination with a robot capable to modify it is actions accordingly. This cannot be achieved with robots programmed by means of guiding. When a robot programming language is used it is in principle possible to recalculate the trajectory. Unfortunately most current implementations of robot programming language are so slow that such a recalculation would drastically increase the idle time of the robot. This is mainly caused by the use of interpreters instead of compilers.

5.4. The design of a programming interface for computer controlled machinery

The programming and reprogramming of the FMS should be made sufficiently easy that it will not become a bottleneck in the development of a new product. When the complexity of FMS's increases it is questionable whether current programming techniques will satisfy this requirement. An important handicap is the wide variety of programming interfaces. An individual company may try to solve this by using equipment of only one vendor. If such a strategy would get widespread use it will most likely become a serious danger for the development of a competitive European industry of FMS suppliers. To prevent this it is necessary that suppliers of machinery used in FMS's will take care to adhere to standard approaches and will resist the temptation to profile their products by offering "new" and "special" programming systems.

Currently most programming of robots is done on the factory floor whereas numerical controlled machines are using elsewhere generated programs. It is to be expected that in the future robots will be programmed on a remote system as well, since program development can then take place without using the machine the program is intended for. The disadvantage that no direct visual control of the program is possible has then to be compensated by simulation systems. As a remote development system a personal workstation with a graphics display can be used. It is possible in small applications of FMS that the same computer system is used as development and as control system. Developing programs on an other system requires that the necessary programming tools for the computer controlled machine are also available there. There are two possible choices for the interface between the program development system and the computer controlled machine.

1 It is possible to have the high level program transferred to, or developed on, the computer controlled machine. In that case on the computer controlled machine either a compiler or an interpreter for the high level program has to be available.

2 The alternative is to have a compiler for the high level program on a separate development system translate the program and to transfer the translated code to the computer controlled machine.

Currently the first alternative is the one most widely used. The main advantage is that the computer controlled machine can operate as a stand alone system. In a CIM environment the second strategy is to be preferred. The hardware used in the computer controlled machine can be simpler (and therefore cheaper) since it doesn't need to support a compiler or interpreter. This becomes important when many computer controlled machines are used or when complex compilers are used which require the support of a time-sharing system. A second advantage is that during programming the machine for which the program is inteded doesn't have to be in an off-line status as is currently the case.

For programming computer controlled machinery general purpose programming languages, like Fortran, Basic or Pascal, are not suitable. On the other hand practically all features of a general purpose programming language are also needed when programming a computer controlled machine. Procedures, conditional clause, repetitional clauses, input and output are necessary. This implies that a programming language for computer controlled machinery should be an extension of a general purpose programming language. It is important that a standardized programming interface with computer controlled machinery comes into existence. The development at Boeing of an APT based robot programming language indicates that it is indeed possible to have a single programming interface for all computer controlled machinery. The programming language APT, now widely used for programming NC machinery, is however not suitable to become the basis of a standardized programming interface for computer controlled machinery because of its cumbersome and antique syntax.

The required extensions to an ordinary programming language can be grouped as follows:

- Additional facilities for communication with other processors.

A program for a computer controlled machine should be capable to communicate with a program in the central control to report errors and statistics and to receive orders. Also it needs to communicate with the control program of an other machine in the same unit.

- Facilities to handle asynchronous events

The Pascal successors Modula-2 and Ada both give facilities to respond to asynchronous events. Unfortunately these facilities are each of a different nature and it is not to be expected that either solution will acquire general acceptance in the near future. A second problem is that the facilities to handle asynchronous events are probably of a too low level to be easily used in a program for a computer controlled machine. What is needed are constructs which are directly coupled to the way asynchronous events are handled. In existing robot programming languages constructs of the form "perform a certain action, until a certain event happens" have been introduced. It is not clear whether such extensions should be in the form of procedures that can be called by the user , or in the form of a special syntax.

- Machine dependent commandos

Every computer controlled machine will have typical capabilities to be controlled by its program. Some of these capabilities will be typical for a class of machinery, some just for one machine. For robot programming, for example, one needs commands to open a gripper, to close it, check whether it has grasped something, etc.

- extensions for easy definition of curves and surfaces in space.

The specification of a path that should be followed by a cutting tool or a robot is essential for a programming language for computer controlled machinery. Facilities for doing so are the most important part of the language APT for NC machines. This is also important for robots since one of the major problem areas there is the specification of a motion in space, and in what coordinate frame it should be specified.

- An interface with the data produced by CAD systems

In an computer integrated manufacturing environment it is necessary that the programmer of the computer controlled machinery can make use of the data produced by the CAD system.

It is not desirable that every language for computer controlled machinery is based upon a different programming language, especially since there is a programming language, Pascal, which has become a world-wide accepted starting point for further development of programming languages. For the immediate future we don't think that it is realistic to expect that a standard programming language for the programming of computer controlled machinery can be constructed. To avoid an unnecessary divergence of the languages produced by each manufacturer we specify that:

- **A programming language for computer controlled machinery should be based on a well structured high level language such as Pascal or Ada.**

  Such a language has the following characteristics:

- Strong typing;
- The control structures such as they are used in Pascal (If - then - else, while - do, recursive procedures);

5.5. A Processing Strategy for FMS

In FMS their are two types of processing:

- The processing for the machines actually modifying or assembling the workpiece;
- The processing for the routing of the workpiece between FMS cells.

The processing at the FMC consists of real-time control and sensor processing. Also a local man-machine interface with graphical display is necessary, even when programming is mostly done on a remote system, to make on the floor corrections and to intervene in case of malfunctioning. The basic processing strategy for the FMC is to make them sufficiently independent so that they don't need to rely on external computing resources. The primary advantages of this strategy are that:

- Each unit can be (further) automated independently of the others. This is an advantage that only counts when it is either economically or technically not feasible to automate the production system in it's entirety.
- Communication can be, and in most cases is, restricted to receiving task level commands from a central computer, including information about transformation of parts and tools. The cell only reports back about status of tools and parts to the central computer.
- Failure in one unit will not affect other units directly;

Within the FMC a hierarchical structure of processes is needed to achieve modularity and to make it possible to use more then one cpu. This processing strategy is made possible by the ongoing revolution in micro electronics, and especially the field of microprocessors. Today even the largest mini computers are equaled in processing speed by the newest generation of microprocessors. Only very large mainframes and vector processors offer significantly more processing power. Nevertheless it is possible that a unit needs more computing power then is offered by a single "micro" system. This cannot be solved though by a shared large processing facility since those computers capable to do essentially more processing then micro systems are not capable to guarantee a fast response in real-time. What is needed instead is the possibility to use a number of micro processors.

A single master computer for the entire FMS is needed for control over the type of products and the rate of production and to respond to error messages from the various machinery employed.

The routing of the workpieces can either be fixed for a particular product, or flexible to optimise for machine usage. If the routing is fixed for a particular product it is sufficient that the central control gives the appropriate command to the flexible transport system at the start of a new batch. When attempts are being made to optimise machine usage and depending on the situation of the moment a workpiece can be transferred to machine A or B, it is necessary that a central process tells every time a workpiece arrives at a crossing point where to it should be routed. This can, depending on the cycle times of the machines and the amount of crossing points, result in a large processing demand. It might therefore be necessary to use a separate computer system or even a number of them for this task.

5.6. A Communication Strategy for FMS

The consequences of the chosen processing strategy for the communication strategy are that:

- The FMS is divided in independent units which don't communicate directly;
- Within a unit a sophisticated communication facility between the processing units is necessary to allow for fast real-time cooperation.
- All units should be capable to communicate with a single central control.

The communication needed between computer controlled machines within a unit is of a special nature. When, for example, two robots are used to lift one object precise synchronisation is needed to ensure that neither of them gets too much of the weight to carry. For this type of communication where the time that a message may travel is of the order of a millisecond it might be that even a fast serial link is too slow and that a shared memory is necessary for communication. Alternatively the same techniques as they are used for local area networks might be employed but then this local area network should be local to the machines in the unit and no one from outside should be capable to directly access this network. This requirement is necessary to ensure that the real-time characteristics of the network within the unit would not be disturbed.

The communication between the central control and the machining units will be limited to the transfer about error rates from the units to the central control and the transfer of commands from central control to the units about the type and number of products to be made. Consequently only a simple communication interface between the units and the central control is needed. Nevertheless it is advisable to make use of a local area network since the effort needed to make a simple special purpose interface might well be more expensive then to buy a standard local area network. Secondly considerable less interface and integration problems will arise when use is being made of a local area network for communication between the central control and the units.

When a routing scheme is used that determines during run time on which machine a product will have to be machined the communication load depends on the number of workpieces which arrive at a crossing per unit time. If this number is of the order of one to ten per second a single local network will be sufficient to handle the communication and in that case the same network can be used as for the connection of the central control process with the machining units. If the number of routing decisions is well over hundred per second current local area network techniques make it impossible to use one network and an hierarchical approach becomes obligatory.

Apart from the communication required during production time there will in an integrated computer manufacturing environment also be a need for communication during program development. This communication need is caused by the necessary transfer from information originating in the computer aided design system about the product and the production process, and by the transfer of programs not developed on the factory floor to the computer controlled machinery.

As far as the interface between the central control of the flexible manufacturing system and other CIM sub systems is concerned a communication strategy has to be based on the need of these other sub systems to communicate.

References

Bryce, 1982. Bryce, A. L. Graham and Roberts, P. A., "Flexible Machining Systems in the USA," in *Flexible Manufacturing Systems, Proceedings of the 1st International Conference, Brighton, UK, 20-22 october 1982.*, IFS (publications) Ltd., North Holland Publishing Co (1982).

Lozano-Perez, 1982. Lozano-Perez, T., "Robot Programming," A. I. memo No. 698, Massachusetts Institute of Technology Artificial Intelligence Laboratory (December 1982).

Ohmi, 1982. Ohmi, T., Ito, Y., and Yoshida, Y., "Flexible Manufacturing Systems in Japan - Present status -," in *Flexible Manufacturing Systems, Proceedings of the 1st International Conference, Brighton, UK, 20-22 october 1982.*, IFS (publications) Ltd., North Holland Publishing Co (1982).

Spur, 1982. Spur, G. and Mertins, K., "Flexible Manufacturings Systems in Germany, conditions and development trends," in *Flexible Manufacturing Systems, Proceedings of the 1st International Conference, Brighton, UK, 20-22 october 1982.*, IFS (publications) Ltd., North Holland Publishing Co (1982).

6. Computer Aided Design of Solid objects.

6.1. Basic concepts of Computer Graphics.

Computer Aided Design (CAD) is the application area of Data Processing, which provides computing assistance to a designing engineer. A dominant feature of any CAD system is the support for the creation and manipulation of pictures (Newman, 1979). So defined, CAD can be viewed as a subfield of a wider area, that of Computer Graphics Applications.

Among the fundamental concepts of Computer Graphics the following topics can be identified:

- Coordinate systems
- Line drawing
- Two dimensional transformations, clipping and windowing
- Output device handling (plotters, display devices, raster scan displays etc.)
- Graphics subroutine packages
- Three dimensional transformations and perspective
- Input device handling (mouse, tablet, lightpen)
- Representation of curves and surfaces
- Modelling of three dimensional objects
- Hidden surface elimination
- Raster graphics, colouring and shading
- Graphics systems and user interface design

Even a simple graphics system will already encompass the majority of these concepts. For most of these topics well-established design rules exist.

As an example, due to Newmann and Sproull, (Newman, 1979) algorithms on "Line drawing" using point plotting techniques should generate lines which have (at least) the following properties:

- Lines should appear straight and not crooked. Point plotting techniques are suited for lines at 0, 45 and 90 degrees from the horizontal edge of a display, but at other angles a line might pass through non-addressable points and must be approximated by choosing addressable points close to it in such a way, that the line appears to be straight instead of crooked (figure 6.1)
- Lines should have constant density. The line density, which determines the brightness (or blackness, if the line is dark) is proportional to the number of dots displayed divided by the length of the line. To maintain constant density, the dots should be equally spaced. This requirement can be achieved only in parallel lines or lines at 45 degrees to the edges of the screen. The algorithm should attempt to achieve an as even spacing as possible, otherwise bunching of dots will result in bright or dark spots on the line.

Furthermore, it is obvious, that

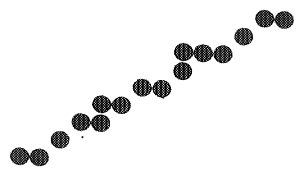

Figure 6.1. Crooked line

- Line density should be independent of line length and angle.
- Lines should be drawn as rapidly as possible in interactive applications, ideally by special purpose hardware.

The latter requirement can be formulated for any topic listed, when the resulting system has to be used in an interactive environment, as most today's systems are.

Although this simple example is not particularly relevant for CAD, it should be noted that such well-established rules exist for almost every topic in Computer Graphics listed above and any CAD system, which is not properly designed according to the rules of the basic techniques used in Computer Graphics is not likely to be very successful. Hence, a general design rule for all Computer Graphics applications, including CAD, can be formulated.

Any Graphics Application System should be designed according to well-established rules developed in the area of Computer Graphics during the past twenty years.

These rules and guide-lines can be found in textbooks, e.g. in (Newman, 1979, Foley, 1982).

6.2. Use of standards

In addition, a number of topics listed have reached a state of maturity making them suitable candidates for **standardization.** In particular, this is the case for topics such as:

- Use of coordinate systems;
- Two dimensional transformations, clipping and windowing;
- Graphics subroutine packages;
- Display files and segments.

An important ISO-approved standard, covering these area's, is the Graphics Kernel Standard (GKS) (ISO, 1982), which can be viewed as a solid base for future Computer Graphics applications (Hopgood, 1983). Implementations of GKS are under development or have been completed at the time of this writing. In addition, an extension of this standard is currently being developed to include three dimensional transformations, perspective, clipping and windowing.

The strongest justification for standardization is to achieve program portability, i.e. the ability to run programs on different computers with different hardware configurations while changing the program as little as possible, ideally without modifying it at all (Newman, 1978). Therefore, programs, that are designed using widely accepted appropriate standards to the extent possible, are usable on far more computer installations than programs that are not. As a consequence, they have a bigger market.

Furthermore, using high-level standardized functions like, for example, those provided for by GKS, the designer of a graphics system or application program is not required to conceive and implement these functions himself; it is only necessary that these functions are thoroughly understood. Hence, using an implementation of GKS, the task of an implementor of a graphics system or the graphics part of an application program, is relieved, which reduces the cost of an actual implementation.

Finally, since an effective standard is thoroughly understood by many people and especially when good tutorial textbooks exist, which is the case for GKS (Hopgood, 1983, Encarnacao, 1983a), the training problem of programmers is reduced and it becomes possible to achieve a fairly high degree of "programmer portability" (Newman, 1978).

As a drawback of the use of standard software it can be argued, that this software is not able to optimally exploit available hardware facilities or that such software is not fast enough to meet real-time requirements, which are very important in an interactive graphics system.

Indeed, it is evident, that a standard should be applicable and span a wide class of existing computer hardware and peripherals. Also, there should be possibilities to access unusual hardware features and incorporate future technological innovations. GKS meets these requirements to a large extent.

With respect to the efficiency of a software implementation of a standard, which is to be shared between many costly and demanding application systems, one can state, that it should be designed and implemented with utmost care, not only for consistency, completeness and robustness, but also for performance and economy by selecting the best modularization, data structures and algorithms feasible.

Whenever such an implementation of a standard exists, it is judged to be unlikely that an ad-hoc implementation of similar functions as provided for by standardized software would result in better performance than using the implementation of that standard.

Furthermore, it is technically feasible and economically justifiable, that the computational expensive parts of an implementation may be performed by special purpose VLSI hardware, e.g. the GKS-chip, which is currently being designed (Encarnacao, 1983b). Such implementations, in addition to modern microprocessor technology, combine high efficiency with a high level graphics interface allowing for portability **and** increasing performance.

Based on these arguments, the following design rule may be formulated:

Whenever suitable international accepted standards for computer hard- and/or software systems and carefully designed implementations of these standards exist, they should be incorporated in the design, rather than constructing modules to perform functions as provided for by these standards.

6.3. Geometric modelling systems

Many of the basic elements of Computer Graphics, discussed in the previous section, play an important role in the design and implementation of a large number of application systems. These applications may vary from simple data display programs (e.g. well-known profit versus time pictures) to powerful and elaborate systems like flight simulators and sophisticated CAD workstations. Concentrating on the latter and, more specifically, on those CAD systems appropriate for the design of discrete parts, CAD systems aimed at other purposes, e.g. VLSI-design or architectural design will be excluded from the following discussion.

In recent years, a number of powerful systems have been developed specifically suited for the design of three dimensional solid objects known as "Solid Modellers" (Requicha, 1982, Requicha, 1983b). These systems originate from some academic sites, e.g. Build-2 (Hillyard, 1982) and PADL-1/2 (Brown, 1982) and from airplane and automobile industries, e.g. GMSolid (Boyse, 1982) The latter system is a direct offspring of the PADL-system developed at the University of Rochester.

The term **Solid Modelling** denotes a wealth of theories, techniques and systems aimed at constructing **informationally complete** representations of solids. Such representations enable, at least in principle, that any well-defined geometrical property of any represented object can be calculated automatically.

This is important, since the geometrical properties of a designed part, especially its shape, play an important part in any design phase. Usually a designer starts with a shape definition. The identification of sub-components can be done easiest by referring to the geometry. Also, non-geometrical data (such as materials, strength, tolerances, etc.) need the geometrical model for the identification of the component they apply to. The production designer refers to the same geometry for specifying machining and assembly operations.

Therefore, it can be argued, that the **same** geometric model is the base for design support in almost every design. Hence, the capabilities of the geometrical modelling methods determine to a large extent the degree of integration possible for the components of the Computer Integrated Manufacturing (CIM) system.

Increasing the degree of integration may result in a far higher level of flexibility in industrial automation, so that in course of time the "A" in CAD/CAM might stand for "automated" rather than "assisted".

6.4. Representation Schemes used in Geometric Modelling.

Since a geometric model of an object is the coupling between CAD (where it is created), CAE (where it is analyzed) and CAM (where it determines the final result of the manufacturing process), geometric modelling must be regarded as a vital module in terms of Information Technology with respect to Computer Integrated Manufacturing. Within geometric modelling, the representation schemes available for representing solid objects can be regarded as a key issue, since their properties determine the feasibility to automate activities needed in CAD, CAE and CAM and the associated processing characteristics.

To justify this, the common representation schemes presently used in geometric modelling will be explained. Moreover, it will be shown, that particular representation schemes are advantageous or even necessary in particular application areas. Hence, during the course of the whole CAD/CAE/CAM process, representation conversions are needed at carefully selected points in that process.

When discussing the properties of representation schemes it is advantageous to distinguish those properties, which can naturally be *formalized* from those, for which that cannot easily be done. When the former are adequately and consistently defined, it will be possible to make unambiguous and verifiable statements. To show, how this can be achieved, some basic notions and definitions will be discussed.

Since the material discussed in the following paragraphs is complicated and based on several extensive mathematical theories, the treatment as given here is necessarily rather superficial. For detailed information, the reader is referred to textbooks, e.g (Encarnacao, 1980, Nowacki, 1982, Encarnacao, 1983a). and reviews, e.g (Requicha, 1980, Requicha, 1982, Requicha, 1983b). The discussion, presented here, closely follows the treatment by Requicha (Requicha, 1980).

6.5. Properties of abstract solids

An *abstract solid* can be described as a pure mathematical notion, which should have the following properties in order to be of practical use:

(1) *Rigidity.* An abstract solid should have an invariant shape, which is independent of its position and orientation in space.

(2) *Homogeneous three dimensionality.* An abstract solid must have a well-defined interior and a solid's boundary cannot have isolated or "dangling" portions (figure 6.2).

(3) *Finiteness.* A solid must occupy a finite portion of space.

(4) *Closure* under rigid motions and a number of Boolean operations. Translation and/or rotations or operations that add or remove material should, when applied to solids, produce other solids.

(5) *Finite describability.* Solids must have distinct finite aspects (e.g. a finite number of faces) in order to be representable in a form, suitable for computer processing.

(6) *Boundary determinism.* The boundary of a solid must determine unambiguously what is "inside", and hence comprises the solid.

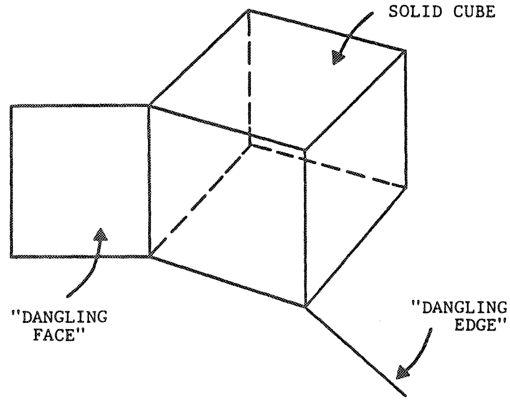

Figure 6.2. "Dangling" face and edge.

6.6. Formal definition of representation schemes

In order to obtain mathematical models having the properties listed in the preceding section, it has been shown (Requicha, 1977), that suitable models are (congruence classes of) subsets of three dimensional Euclidean space which must be bounded, closed, regular and semianalytic. Requicha calls these sets **r-sets** (regularized sets). These mathematical notions are well-defined (Kuratowski, 1976), and may be explained intuitively. The set in figure 6.2 is closed because it contains its boundary, but it is not a regular set because of the existence of "dangling" portions, which do not define an interior. Such "dangling" portions could easily result from Boolean operations. Figure 6.3b contains a two dimensional example. This example also shows, that r-sets are not closed under conventional set operations, since the set depicted in figure 6.3b is not an r-set, because a line segment does not determine an area.

However, Tilove and Requicha have shown, that, without loss of mathematical rigour, one may define a **regularization** operator, which, when applied after each of the Boolean operations taking union, intersection, difference or complement, result in a regular set. Informally, regularization can be described as: take everything inside the set, excluding its boundary, and cover this with a "skin" to form a closed set (Tilove, 1980). More formally, one may define the *regularization* of a set X as $r(X) = ki(X)$ where i and k denote the interior and the closure of a point set according to the conventional definitions (Kuratowski, 1976). In this way one may remove effectively point sets such as the "dangling" line segment in figure 6.3b to obtain the regular set in figure 6.3c.

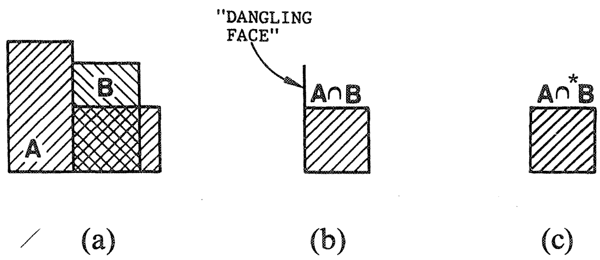

Figure 6.3. Regularization after a set operation

One may now postulate a *mathematical modelling space M* whose elements are the r-sets, which can be seen as mathematical models of abstract solids.
Further, one may define syntactically correct representations as finite symbol structures constructed with symbols from an alphabet according to syntactical rules. The collection of all syntactically correct representations is called a *representation space R*.

Now, a representation scheme is defined formally as a relation $s: M \rightarrow R$ with domain $D \subset M$ and value range $V \subset R$ (figure 6.4). Any representation r in V is said to be **valid** since it is both syntactically (it belongs to R) and semantically correct (it has one or more elements in domain D). A representation r in V is said to be **unambiguous** (or **complete**) if it corresponds to a single-element subset of D. It is called **unique** if its corresponding objects in D do not admit representations other than r in V. A **representation scheme** s is unambiguous if all of its valid representations are unambiguous and, similarly, it is unique if all of its valid representations are unique. Note, that unambiguity of a representation neither implies nor is implied by its uniqueness. Informally, this means that a valid representation is ambiguous if it corresponds to several solids and a solid has non-unique representations if it can be represented in several ways by means of a particular representation scheme.

6.7. Formal properties of representation schemes

The definitions of the preceding section have some important practical implications for the representation schemes, which are being used in actual Geometric Modelling Systems (GMS).

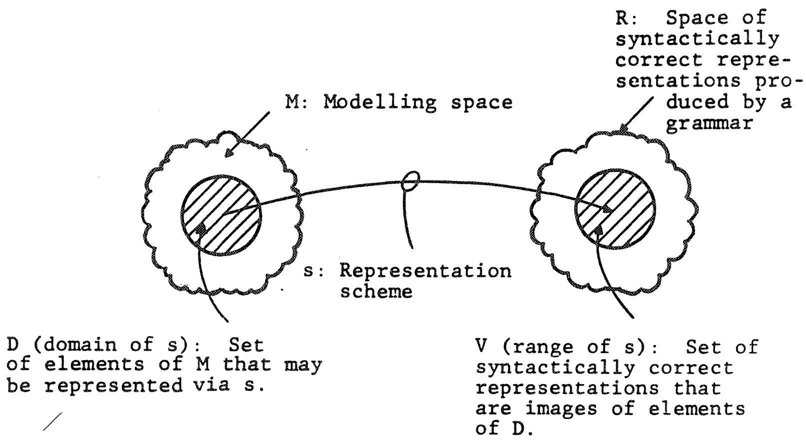

Figure 6.4. Domain and range of a representation scheme

6.7.1. Domain

The domain of a representation scheme is the set of entities, that can be represented by the scheme and therefore characterizes its **descriptive power.**

6.7.2. Validity

The range of a representation scheme is the set of representations which are valid. It is obvious, that **representational validity** is very important to ensure the integrity of the databases on which an actual GMS is operating. Those databases should not contain nonsense objects, otherwise such invalid information could result in a system crash, obvious suspect results, or, in the worst case, apparently credible results in fact being meaningless and worthless.

In the latter case, it may take considerable time before the invalidity of the result is determined so that the fault might be propagated in an unpredictable way through the whole CAD/CAM process. This must be considered very dangerous and should be precluded at all cost.

Since ensuring validity of representations by humans, e.g. by inspecting a drawing on a graphics display, is error-prone or hardly possible, it is essential to check validity by automatic means. The most straightforward way to achieve this is using representation schemes in which all syntactically correct representations are valid (e.g. Constructive Solid Geometry). This reduces validity checking to a parsing problem.

6.7.3. Unambiguity

Engineering drawings, the traditional means for specifying solids, can be viewed as informal means of communication among humans. Because engineers and technicians possess a vast amount of "world" knowledge, they usually interpret these drawings correctly but sometimes they make errors or, inversely, they interpret incorrect drawings as if they were correctly, perhaps without even noticing the error. Automated machines, e.g. programs that interpret drawings, unfortunately do not have these capabilities in general and therefore, unambiguous representation schemes are mandatory.

Until recently, commercially available geometric modellers were mostly based on ambiguous so-called "wire-frame" representations (Voelcker, 1977).

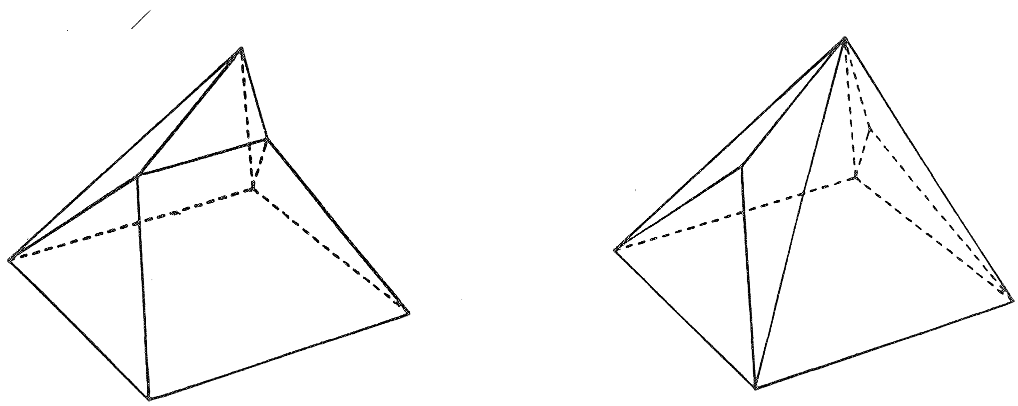

Figure 6.5. A polyhedron is not uniquely defined by a list of vertices.

For example, a polyhedron, when represented by a list of vertices, is not uniquely defined in such a scheme, as can be seen from figure 6.5. Even a list of edges does not represent an unique polyhedron (figure 6.6).

Therefore, a necessary condition for the automatic integration of modellers and other automated systems, is, that object information should be represented according to an unambiguous representation scheme. As shall be shown later, such representation schemes exist and are successfully implemented in recently developed practical modelling systems known as **"solid modellers"** . Some authors, e.g. on page 240 of (Encarnacao, 1983a) point out, that simple but ambiguous representation schemes like wire-frame models have useful applications. For example, the relative simplicity of the algorithms make such schemes suitable for hardware implementation, so that the model or the viewing point can be changed in real-time, as perceived by the designer, enabling him to investigate many design variants or aspects of the design in a short time.

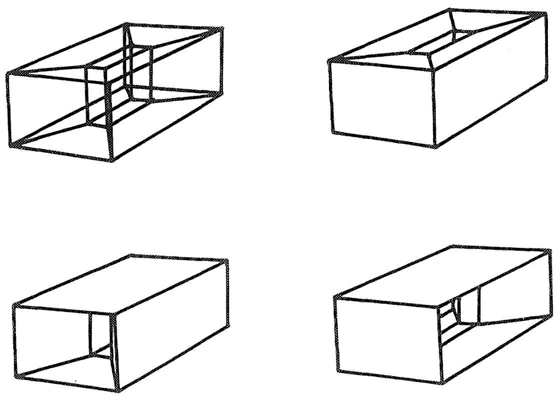

Figure 6.6. Ambiguity of a wireframe representation.

However, since it is quite possible to construct a wire-frame representation from an exact representation automatically (Requicha, 1983b), while the reverse is not true, it seems more appropriate to store the complete model using an exact representation. Such a general, as complete as possible representation should be the starting point for extracting additional representations which, although no longer unambiguous, are better suited for particular kinds of processing (e.g. real-time display).

6.7.4. Uniqueness

The uniqueness of a representation scheme is an important property for establishing the equality of objects. For example, programs for Numerically Controlled machining tools must eventually produce objects equal to those specified by product designers.
As another example, planning programs, that search through a space of alternatives, may loop indefinitely if previous situations cannot be recognized. In practice, most representation schemes are not unique for at least the following reasons:

(1) substructures in a representation may be permuted

(2) different representations may correspond to differently positioned but congruent copies of a single geometric entity

While both cases are conceptually easy to distinguish, determining whether two structures contain the same elements may be computationally expensive (especially when these structures are large) and the design of algorithms, that decide whether two geometric entities are congruent is a complicated task.

From the above discussion, it is apparent that the primary representation of a model should be both valid, unambiguous and unique.

When a justifiable need exists in a particular application, representation conversions can be used to obtain a representation, that is suitable to perform a particular function on that model.

6.8. Informal properties of representation schemes

Besides the above properties, which can be formalized, as shown, there are also a number of important properties, which cannot be easily formalized in an adequate way.

6.8.1. Compactness

Representation schemes, which naturally result in compact data structures, have obvious advantages, since they can easily be stored or transmitted over data links. However, compactness should not be aimed at to the extreme. Carefully chosen redundancy may have important advantages such as:

- wider applicability
- informationally completeness
- possibility to easily detect and correct syntactic errors in representations
- often dramatical improvement of computational speed by storing rather than computing repeatedly needed data.

It is often difficult to predict, whether or not data storing is to be preferred over repeated computing. This may depend heavily on a particular application or on a particular situation created by the user. A better strategy would be to determine dynamically (i.e. during run-time) which approach is most desirable, e.g. by measuring the real-time spent in a particular computation over the same dataset.

More generally, a good general purpose CAD system has to decide dynamically about a great number of optimum configurational parameters and the data management must, more or less independent of the representation scheme used, allow for dynamic extensibility whenever it is evidently advantageous to use less concise information schemes (dynamic memory allocation).

6.8.2. Ease of use

Since CAD systems should be characterized as *interactive systems,* where a software system interacts heavily with a human being to accomplish a complex task, the interface with the user is of crucial importance. Simple and concise representations are generally easier to create and to manipulate than extended ones, but the former may not convey enough adequate information. Using an extensive representation scheme will be a tedious and error-prone job unless the designer is assisted by elaborate input mechanisms.

Moreover, such modelling systems should also contain mechanisms to ensure the validity and consistency at any significant stage in the design process. It is quite feasible, that knowledge-based analyzing systems may become of great help for the designer in the near future.

6.9. Important representation schemes used in geometric modelling

6.9.1. Wireframes

The first useful representation schemes are the wireframe representations. They first appeared as simple, interactive 2-D systems for the design of printed circuit boards and for 2-D mechanical drawings. The internal representations used were mostly simple lists of lines and arcs. Later, in the 1970's, they were extended to lists of segments of 3-D curves. These can be computed so that it is possible to obtain orthographic, isometric and perspective views. Although 3-D wireframe systems are clearly useful, they have some serious deficits. In figure 6.6 is illustrated, that a wireframe is ambiguous, since it may represent any of the three shaded objects. Other examples are given by Markovsky and Wesley (Wesley, 1980). Furthermore, a wireframe systems may tolerate nonsense objects and there is no possibility to check the validity of the generated representations. Finally, they are not concise and users have to supply a lot of low-level data to describe something as simple as a cube.

6.9.2. Boundary representations

Boundary representations are those schemes, that represent a solid in terms of its boundary or enclosing surface. These boundaries are usually represented as the union of *faces,* while each face is defined in terms of its boundary (normally a union of edges) and additional information defining the surface in which the face lies (figure 6.7).

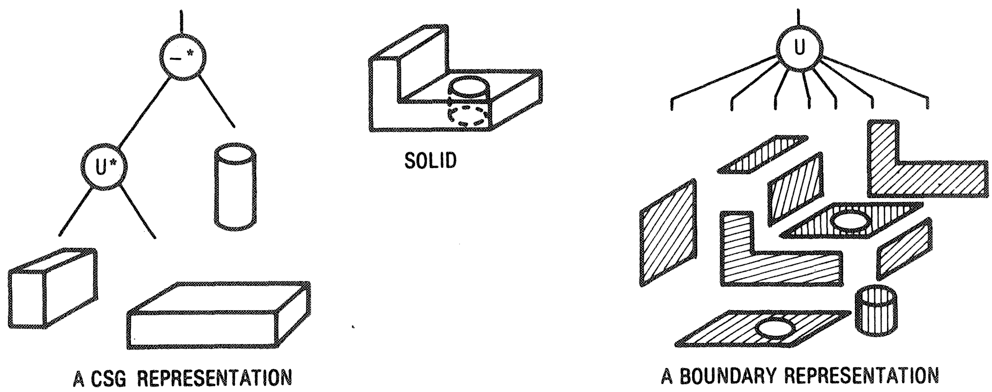

Figure 6.7. A CSG and a boundary representation.

If these faces are represented unambiguously, boundary representations are, in contrast to wireframes, also unambiguous. This follows from mathematical theorems, proving that regularized sets are unambiguously defined by their boundaries (Requicha, 1977). Furthermore, under an extensive number of conditions, which are not easy to verify in practice, they can be guaranteed to be valid. However, in general, they are not unique.

6.9.3. Constructive solid geometry (CSG)

Constructive Solid Geometry (CSG) denotes a family of representation schemes, where complicated solid objects may be represented by ordered additions, subtractions or regularized Boolean operations of simpler solids. The simplest objects of such a scheme (e.g. blocks and cylinders) are usually bounded primitive solids.

These schemes are unambiguous and, moreover, the algebraic properties of regularized sets guarantee that any CSG-tree is valid if the primitive leaves (i.e. the primitive solids) are valid.

However, in general, CSG representations are not unique. The domain of a CSG scheme is largely determined by the primitive objects and the operations supported. Generally, they are an order of magnitude more concise than boundary representations and validity checking is almost trivial compared with the latter.

Boundary representations, on the other hand, have advantages for generating line drawings and graphic displays, since the important data for these purposes are face, edges and the relations between them.

It may be quite laborious to construct a boundary representation because of the vast amount of data to be specified by the user. In a number of actual geometric modellers, CSG representations are used as an input technique and the data thus generated are subsequently converted to a boundary representation because of its advantages with respect to the processing necessary for interactive display.

6.10. Research directions and future trends.

Although the solid modelling approach is to be regarded as a significant step forwards when compared to the classic wireframe modellers, neither of the representation schemes discussed are completely satisfactory. Moreover, there are a large number of open issues, which are studied intensively at a number of research institutions. Some of them are discussed briefly in the following sections. This material is based on recent papers by Requicha and Voelcker (Requicha, 1982, Requicha, 1983b).

6.10.1. Sculptured surfaces

Since the solid modelling representation schemes discussed only support (possibly complex) combinations of primitive rigid objects, which have a simple and regular structure, they are not well suited for the representation of objects with sculptured surfaces. Such objects are often adequately described using Bezier curves or B-spline techniques. It is desirable to incorporate these in solid modellers and to obtain good algorithms to support Boolean operations, mass property calculation, interference analysis and other capabilities, normally found in solid modellers, also for sculptured solid objects.

6.10.2. Representation of tolerancing information.

For many design and production activities it is essential to have facilities for the representation of tolerancing information. Contemporary solid modellers do not possess such facilities, unfortunately, but studies on this subject are being undertaken at various places (Requicha, 1983a, Requicha, 1983c). Also, representation of stressed objects is beyond the scope of solid modelling, but highly desirable from an engineering point of view (Myers, 1982).

6.10.3. Improving computational speed.

Although presently available solid modellers support high-level and very powerful operations, they are often perceived by users as being relatively slow. Both improving algorithms (e.g. conversion algorithms between various representation schemes) and the development of special purpose hardware are obvious methods to attack this problem. An example of the latter are octree representations , which can be used to approximate solids for purposes of display and the generation of meshes for analysis by Finite Element Methods. Because of the regular structure of such representations, they are suitable candidates for hardware implementations (Meagher, 1982).

6.10.4. Applications

Since solid models are unambiguous it is possible, in principle, to support fully automatic algorithms for any geometric application. However, for most application areas, these algorithms are still under development. A few examples will be given.

Interference analysis.

Static interference checking can be defined as follows: if *A* is a collection (e.g. an assembly) of solids *S1, S2, ... ,Sn* then take all pairwise regularized intersections of the solids in *A*. If all these intersections are empty (i.e. they all generate the null-object), they do not interfere. Thus, a system is able to perform static interference checking if

(a) it is able to represent objects as regularized Boolean compositions

(b) it is able to compare represented objects with the null-object (Boyse, 1979).

Dynamic interference checking, where components are swept through spatial trajectories, is a much harder computational problem and requires the ability to detect a null-object in four dimensional space-time (Esterling, 1983).

Graphics display.

While all modelling systems have features for displaying objects being designed, techniques are improving, e.g. to generate realistic illuminated and/or shaded displays using ray-casting techniques, e.g. (Cook, 1982), or to generate pictures with real-time motions.

Finite Element Mesh generation.

The Finite Element Method (FEM) is one of the most important engineering tools for the analysis of mechanical structures, that exist today. Normally, FEM-programs operate on collections of geometrically simple elements, like cubes or tetrahedra. These collections are commonly called "meshes", and are different from the representations used by Geometric Modelling Systems. Since it is a tedious and error-prone job to generate these meshes manually, it is very desirable automate this task and various algorithms to achieve this have been reported, e.g (Wordenweber, 1981).

Numerical Control.

The automatic generation of Numerical Control programs from geometric models is a complicated problem, especially when multiprocessing programs for machining centers have to be generated. Several investigations in this field have been undertaken, e.g. (Armstrong, 1983)

Simulation.

Usually, solid modellers are restricted to representations of rigid objects and assemblies, but it appears, that they can be extended quite easily to represent moving mechanical objects such as industrial robots, as reported by (Tilove, 1983).

Robotics.

It is evident, that the application of robots in modern industries is vitally important. Today, usually they are being programmed off-line, by teach-in methods. In the near future, however, on-line programming will become essential, and it is being studied in the context of solid modelling (Pickett, 1983). A difficult problem is the automatic generation of robot action plans from high-level task specifications, which involves issues such as path planning for collision avoidance (Lozano-Perez, 1981).

6.11. A processing strategy for Geometric Modelling Systems.

Geometric Modelling Systems (GMS) can be primarily characterized as systems, that put a heavy load on processor capacity (floating point operations). They therefore belong to the number crunching category.

Most basic modelling operations, e.g. union (gluing of parts), intersection (interference checking) increase more than linear with the number of basic elements of the model. This also holds for visualization functions, for instance, hidden line removal.

For complex models the data handling is also an important aspect. Modelling operations require random access to bulk data. This means, that a sophisticated organization of background storage or database access is necessary.

A special, but for CAD and CAE very interesting, case is the ability to incorporate modelling functions in an interactive design system (e.g. the construction phase). In order to provide the required fast response a simplified (e.g. low resolution) model is to be used temporarily. The real, exact model can still be constructed automatically in the background.

Because many of the modelling functions are applying basically to the same operations many times (e.g. transformations, intersection of faces), they are excellent candidates for special purpose hardware realizations. This is currently a very strongly pursued development, which may totally alter the CAD/CAE scene in the next few years.

However, when considering the numerous applications of basically the same geometrical model throughout the design processes in CIM, which make the complete model very complex, mainframe type support, also involving (relational) databases, will be indispensable in the near future.

The reliability of the model must be maintained, again, by computational methods (validity, accuracy, consistency).

Because of the high costs involved in developing a model, reusability of (parts of) a geometrical model is an important economic demand. Hence, sophisticated filing systems are appropriate.

Last but not least, the model will have many application interfaces for extracting and adding data. this can only become a manageable task if uniformity of access methods and sharing of conversion algorithms and other utilities are achieved.

References

Armstrong, 1983. Armstrong, G., "A Study of Automatic Generation of Non-Invasive NC Machine Paths from Geometric Models," PhD dissertation, Dept. of Mechanical Engineering, Univ. of Leeds (Apr. 1983).

Boyse, 1979. Boyse, J.W., "Interference Detection Among Solids and Surfaces," *Comm. ACM* **22**(1) pp. 3-9 (Jan. 1979).

Boyse, 1982. Boyse, J.W. and Gilchrist, J.E., "GMSolid: Interactive Modeling for Design and Analysis of Solids," *IEEE Computer Graphics and Applications* **2**(3) pp. 27-40 (March 1982).

Brown, 1982. Brown, C.M., "PADL-2: A Technical Summary," *IEEE Computer Graphics and Applications* **2**(2) pp. 69-84 (Mar. 1982).

Cook, 1982. Cook, R.L. and Torrance, K.E., "A Reflectance Model for Computer Graphics," *ACM Trans. Graphics* **1**(1) pp. 7-24 (Jan. 1982).

Encarnacao, 1980. (ed.), J. Encarnacao, *Computer Aided Design: Modelling, Systems Engineering, CAD-Systems,* Springer-Verlag, Berlin (1980).

Encarnacao, 1983a. Encarnacao, J. and Schlechtendahl, E.G., *Computer Aided Design (Fundamentals ans System Architectures),* Springer-Verlag, Berlin (1983).

Esterling, 1983. Esterling, D.M. and Rosendale, J. Van, "An Intersection Algorithm for Moving Parts," *Proc. NASA Symp. Computer-Aided Geometry Modeling,* pp. 129-133 (Apr. 20-22, 1983).

Hillyard, 1982. Hillyard, R.H., "The Build Group of Solid Modelers," *IEEE Computer Graphics and Applications* **2**(3) pp. 43-52 (March 1982).

Hopgood, 1983. Hopgood, F.R.A., Duce, D.A., Gallop, J.R., and Sutcliffe, D., *Introduction to the Graphical Kernel System (GKS),* Academic Press, London (1983).

ISO, 1982. ISO, TC97/SC5/WG2.N117., *Draft International Standard ISO/DP 7942; Information Processing; Graphical Kernel System (GKS).* 1982.

Kuratowski, 1976. Kuratowski, K. and Mostowski, A., *Set Theory,* North-Holland Pub. Cy., Amsterdam (1976).

Lozano-Perez, 1981. Lozano-Perez, T., "Automatic Planning of Manipulator Transfer Movements," *IEEE Transactions on Systems, Man & Cybernetics* **SMC-11** pp. 681-698 (Oct. 1981).

Meagher, 1982. Meagher, D., "Geometric Modeling Using Octree Encoding," *Computer Graphics & Image Processing* **14** pp. 249-270 (June 1982).

Myers, 1982. Myers, W., "An Industrial Perspective on Solid Modeling," *IEEE Computer Graphics & Applications* **2**(2) pp. 86-97 (March 1982).

Newman, 1978. Newman, W.M. and Dam, A. Van, "Recent Efforts Towards Graphics Standardization," *Computing Surveys* **10**(4) pp. 365-380 (Dec. 1978).

Newman, 1979. Newman, W.M. and Sproull, R.F., *Principles of Interactive Computer Graphics (2nd edition),* McGraw-Hill, New York (1979).

Nowacki, 1982. Nowacki, H. and Gnatz, R., *Geometrisches Modellieren,* Springer-Verlag, Berlin (1982).

Pickett, 1983. Pickett, M. and Tilove, R.B., "Robo Teach: An Off-Line Robot Programming System Based in GMSolid," *Proc. General Motors Symp. Solid Modeling,* (Sept. 26-27, 1983).

Requicha, 1980. Requicha, A.A.G., "Representations for Rigid Solids: Theory, Methods and systems," *Computing Surveys* **12** pp. 437-464 (Dec. 1980).

Requicha, 1983a. Requicha, A.A.G., "Representations of Tolerances in Solid Modelling: Issues and Alternative Approaches," in *Proc. General Motors Symp. Solid Modeling,* , Detroit, Mich. (Sept. 26-27, 1983).

Requicha, 1983b. Requicha, A.A.G. and Voelcker, H.B., "Solid Modeling: Current Status and Research Directions," *IEEE Computer Graphics and Applications* **3**(7) pp. 25-37 (Oct. 1983).

Tilove, 1980. Tilove, R.B. and Requicha, A.A.G., "Closure of Boolean operations on geometric entities," *Computer Aided Design* **12**(5) pp. 219-220 (Sept. 1980).

Tilove, 1983. Tilove, R.B., "Extending Solis Modeling Systems for Mechanism Design and Kinematic Simulation," *IEEE Computer Graphics and Applications* **3**(7) pp. 9-19 (May/June 1983).

Wordenweber, 1981. Wordenweber, B., "Automatic Mesh Generation of 2 and 3 Dimension Curvilinear Manifolds," PhD dissertation, Computer Laboratory, Univ. of Cambridge (Nov. 1981).

Encarnacao, 1983b. Encarnacao, J.L., Lindler, R., Mehl, M., Phaff, G., and Strasser, W., "A VLSI Implementation of the Graphics Standard GKS," *Computer Graphics Forum* **2**(2/3) pp. 115-121 (Aug. 1983).

Voelcker, 1977. Voelcker, H.B. and Requicha, A.A.G., "Geometric Modeling of Mechanical Parts and processes," *Computer,* pp. 48-57 (Dec. 1977).

Requicha, 1982. Requicha, A.A.G and Voelcker, H.B., "Solid Modeling: A Historical Summary and Contemporary Assessment," *IEEE Computer Graphics and Applications* **2**(3) pp. 9-24 (March 1982).

Foley, 1982. Foley, J.D. and Dam, A. Van, *Fundamentals of Interactive Computer Graphics,* Addison-Wesley (1982).

Requicha, 1983c. Requicha, A.A.G., "Toward a Theory of Geometric Tolerancing," Tech. Memo. 40, Production Automation Project, Univ. of Rochester (Mar. 1983).

Wesley, 1980. Wesley, M.A. and Markowski, G., "Fleshing out Wire-frames," *IBM J. Research and Development* **24**(5) pp. 582-597 (Sept. 1980).

Requicha, 1977. Requicha, A.A.G., "Mathematical models of rigid solid objects," Techn. Memo. 28, Production Automation Project, Univ. Rochester, Rochester, N.Y. (Nov. 1977).

7. SENSOR SYSTEMS

7.1. Introduction

Sensors are used to resolve uncertainties about the physical state of a system. They can be used to acquire information about relative position and orientation of parts, the shape of parts, touch and force, verification of quality and integrity etc. The sensor itself is a transducer which in most cases produces an electrical signal related to the physical parameter of interest.
This is based on the conversion of one form of energy into electrical energy. If this other form of energy is modulated with some sort of information, the electrical output signal is modulated, carrying the same information. The energy forms that can be converted, can be grouped in six domains, leading to the six signal domains (Lion, 1969) : radiant signals (light intensity or frequency), mechanical signals (pressure, level), thermal signals (temperature, heat flow), electrical signals (voltage), magnetical signals (magnetic field) and chemical signals (PH).

It appears that based on the transduction principle there are two different kinds of sensors. Some of them, called *self-generating sensors* , are sensors which generate an electrical output without an auxiliary source of energy. They directly convert one of the above-mentioned forms of energy into electrical energy. Examples of these are conventional tachometers, thermocouples and solar cells.
The other sort, called *modulating sensors* , are sensors which need an auxiliary (electrical) energy to be able to transform the physical quantity of interest. This transformation is based on the fact that these sensors cause a modulation of the auxiliary electrical energy related to the modulation of the form of energy that is converted. Examples of these are microphones, temperature dependent resistors etc.
It can be noticed that the electrical energy produced by the self-generating sensors is always smaller than the amount of energy put into it. This can be different for modulating sensors, where the energy of the output signal can be higher than the energy of the input signal.

Dependent on the field of interest, there are also other ways to make distinctions between sensors. One way is to make a distinction between *passive* and *active* sensors. The first, as the name implies, are sensors which work with the given environmental condition (e.g. illumination), whereas the latter are sensors consisting of a source and a detector to provide their own source of radiation (for instance sonar systems or structured lighting vision systems). This feature makes active sensors more versatile than passive sensors, since they are less dependent on environmental conditions.

These distinctions all are fundamental for the method of sensing. In general the signal does not indicate by what method it was obtained. On the level of signal interpretation it is even desirable to be shielded off from the details of the sensor hardware.
It is very important to decide which aspects of the sensing signal must be preserved and which aspects should disappear above a certain level of interpretation. It is for instance important to know whether sensors provide *absolute* or *relative* values.

Their possibility to provide *analogue* or *digital* output, can also be a reason to make a distinction, especially useful if their coupling to a read-out or control system can be an analogue or a digital circuit.

Another distinction that can be made is based on the interpretation of the clean sensor signal, which can occur as a *non adaptive* or as an *adaptive* mechanism.
Non-adaptive mechanisms of interpretation use fixed sensors and the interpretation is based on information obtained from a relatively small number of observations.
Adaptive sensing mechanisms utilize the ability to alter the object or the sensor in order to enhance the acquired representation and supplement the information provided. Obviously, to obtain this flexibility, a real-time analysis of the sensor signals is necessary.
For sensor control, the distinction between adaptive and non-adaptive sensors is very important.

In the wide assortment of devices and systems that exist for robot sensing, there is a need to make a distinction between *contact* and *non contact* sensing.
Contact sensing is the domain of **tactile** sensing, defined (Harmon, 1980) as a continuously variable touch sensing over an area within which there is a spatial resolution, and of **force** or **torque** sensing, which is usually a simple vector resultant measurement at a single point.
Non contact sensing is the domain of optical, sonic and magnetic based sensors. The most complex and no doubt very powerful optical sensor is pure vision. Other examples of non contact sensors are beam interrupters, range finders etc.

7.2. Sensor Hierarchy

In general sensor systems can be modelled according to the level structure of fig. 7.1.

Vision
Sensor Complexes and Coordinations
Data Representation
Non Primitive Sensors
Individual Sensors

Fig. 7.1

Individual Sensors.

At the lowest level of the structure are the individual sensors, with their interface to the outside world. Sensors are designed to react dominantly to a specific property. However, often other physical parameters unintendedly contribute to this reaction as well. For that reason the primary signal has to be compensated or filtered to yield a clean sensor signal. This is the main responsibility of this level.
Depending on the nature of the sensor and its task, this clean sensor signal has some external properties such as cycle time, resolution, whether it delivers absolute or relative values etc.

Non Primitive Sensors

At this level, not only the pure sensor signal, but also its context is the field of interest. This is valid in the following three cases:
Sensor arrays , which are sensors of the same sort, grouped in a specific set-up in order to acquire a broad view of the parameter of interest. Examples of this kind are artificial skin (tactile sensor array), vision systems (CCD array's), temperature sensing array's etc.
Directed sensors , in which case not only the pure sensor signal, but also its momentarily (if flexible) orientation is of interest. For instance, in some cases of force sensing in gripper fingers of robots, the desired information is not only the force but also its direction.
Active sensors , where it is vital for the interpretation of the detected information to combine it with certain parameters of the source (in all other cases of active sensors they belong to the lowest level). Examples are computer vision with structured lighting (such as strip lighting), triangulation range finders etc.

Data Representation

The data representation of the non primitive sensor information has to be chosen in such a way that mapping of the relevant information obtained on the data representation is straightforward. This representation has to suppress the peculiarities of the actual sensor used. Depending on the type of sensor, the representation can be in the form of boolean, integer, real, point (x,y) or array (x,y) and combinations thereof.

Sensor Complexes and Coordinations

In most cases there is a need to combine information from different kinds of sensors, to obtain a sufficiently detailed model of the physical state of the system (workspace). For this reason, an interaction between the different sensor systems is desirable. This interaction will have to be coordinated by a controller, which must be capable of:

- requesting the at the time most relevant information.
- reconciling conflicting information.
- recognising concordant information.

It must be possible to define the manipulations in this level as operations on the data representations of the previous level.

Vision Systems

Compared to other types of sensors, vision systems have, in most cases, an order of magnitude higher complexity. For vision systems the lower level processing is not considered. All attention is given to the higher level (complexes and coordinations) and the interpretation level on top of this. Almost all problems and certainly the usefulness of vision systems are associated with this level.
In case the vision output is considered at a lower level, comparable to the other sensors, this is referred to as vision-detection. Vision-detection can be treated together with other types of sensing rather than with pure vision.

7.3. State Of The Art

It is considered useful, not only to discuss the state of the art of sensing restricted to CIM applications, but also to have a look at sensing in the chemical industry. The reason to do this is that the chemical industry has a much longer experience with sensors than the mechanical industry. As a result of this, more advanced sensors are used in chemical industry, for more complex tasks, such as process control. The mechanical industry only recently makes use of sensors for higher level tasks.
The contrary is true when it comes to computer vision. No doubt mechanical industry triggered the development of this most advanced form of sensing. The chemical industry hardly has applications for and experience with computer vision.
The difference between the two industries is that chemical processes usually are continuous, whereas mechanical processes usually are discrete. This is a result of the fact that chemical industry primarily works with fluids, whereas mechanical industry primarily works with solids. This of course has influence on the types and morphological aspects of sensors used in both industries. Nevertheless, mechanical industry can take advantage of the experience obtained by the chemical industry.
A part of sensing in the mechanical industry is the special field of robot sensing. This field of sensing is discussed in a separate paragraph due to the important role robots may have in CIM.

7.3.1. Sensing in Chemical Industry

In the chemical industry, sensors nowadays play a vital role. Their main task is to measure and control the chemical processes. The interesting parameters for this industry are pressure, temperature, flow, level and composition.
The following is a discussion of some methods used to measure these parameters.

7.3.1.1. Pressure

There are several sensor devices to measure pressure. These are in order of their frequency of occurrence: Bourdon, diaphragm, bellow and silicon pressure transducers. All these types of sensors are based on the principle that pressure is translated to displacement of a flexible element. This displacement, which is related with the pressure applied, is in all four types sensed with some sort of displacement transducer. The distinction between the four types is based on their difference in translating the pressure into displacement.

Bourdon. The bourdon pressure transducer is based on the principle that a bended tube will try to straighten itself when an internal pressure is applied. One end of the tube is fixed and the other end is connected to a displacement transducer. The outside of the tube is exposed to atmospheric pressure, which makes this transducer a relative pressure sensor.

Diaphragm. A diaphragm pressure transducer consists of two chambers, separated by a flexible partition, which is connected to a displacement transducer. The first chamber is exposed to the unknown pressure, the second chamber usually is evacuated. In that case the transducer measures the absolute pressure.
The second chamber also can be exposed to a second pressure, in which case this type of transducer can be used for differential pressure measurements.
If the first chamber is omitted, the diaphragm pressure transducer is a dedicated device

for measurement of the atmospheric pressure.

Bellow. A bellow pressure transducer consists of one chamber, the inside of which is exposed to the pressure to be measured. The outside of the chamber is exposed to the atmospheric pressure. The chamber can expand itself in the same sort of way a bellow can. This expansion, measured by a displacement transducer, is made proportional to the applied pressure due to added elastical forces.

Silicon. Silicon pressure transducers usually are of the diaphragm type. The reason to mention them separately is that the above-mentioned transducers are mechanical devices, constructed of several materials. In silicon pressure transducers, the flexible element is integrated with the displacement transducer on one chip, usually combined with a signal processing and temperature compensating circuit.

It is interesting to notice that the frequency of occurrence shows a remarkable relation with the "age" of the transducer type. The older transducer types, are mechanical devices and thus complex, sensitive to wear etc. The remarkable fact is that one would expect that modern silicon transducers by now would have taken their place in a lot of cases, since they are available at low cost, for different applications and able to measure pressures of up to ~35 kPa.
It seems on the contrary, that they still are in the minority, probably due to conservatism and the fact that there is a lot more experience with the older types of transducers.

7.3.1.2. Temperature

Nearly every electrical property of a material or device varies as a function of temperature and could in principle be employed as a temperature sensor. Unfortunately, no material or device has a temperature dependence that over a wide temperature range is a combination of maximal linearity, maximal sensibility and maximal reproducibility. Furthermore properties as size, cost, speed and unassailability can be of importance. For this reason a sensor with the optimal combination of properties has to be selected for each particular application.
Temperature can be measured with temperature dependent resistors, temperature dependent capacitors, piezo-electric crystals, thermocouples, silicon sensors and optical techniques.

Temperature dependent resistors. Thermistors are semi-conductor resistors with a high negative temperature coefficient. Obviously they belong to the so-called modulating sensors. To measure the temperature, these thermistors can be connected to a constant voltage or a constant current supply. Measuring the current flow through the thermistor in the former case or measuring the voltage across the thermistor in the latter case, will give an indication of the resistance of the thermistor and thus of the temperature. In order to measure the correct temperature, care will have to be taken to limit the power dissipation in the thermistor. A high power dissipation could be the result of choosing a too high constant voltage or current, to what one is tempted in order to increase the sensitivity of the sensor.
Thermistors are cheap and available in a wide variety of resistances, shapes and sizes (down to microscopic), which explains for their popularity. A disadvantage is that their temperature dependence is not linear but exponential. This non-linear response can be compensated by several methods, such as employing a logarithmic amplifier, which is a costly method, or the cheap way by simply adding a fixed resistor in parallel, a

linearization method with the disadvantage that the in this way created linear temperature range is limited.
Other popular resistance thermometers are platinum, nickel and tungsten. They have a wide temperature range (platinum from -250 to 1100°C, nickel from -100 to 300°C), have an almost constant positive temperature coefficient and are chemical resistant.
Germanium and carbon resistors are useful at temperatures below 50°K where they exhibit an exponential temperature dependence like thermistors.

Temperature dependent capacitors. The temperature dependence of capacitors can be used to measure the temperature. A common read out method is to let the capacitor be the frequency determining part of an oscillator circuit. In this way the frequency shift of this oscillator becomes a function of the temperature.
Advantages of this method are the simplicity and the fact that frequencies can be easily converted to digital numbers by means of a frequency counter. Disadvantages are that capacitors as thermosensors are not very sensitive, their reproducibility is low (excluding them from exact applications) and further their response-time is long.

Piezo-electric crystals. Another method based on a temperature dependent frequency is making use of specially for this purpose fabricated piezo-electric crystals. These crystals as a part of an oscillator circuit, resonate at a frequency which shifts proportional with temperature.
Advantages of this method are linearity, reproducibility and simple converting to digital numbers. Disadvantages are the large size and cost of the crystal and the cost of the frequency counter, which must, due to the small frequency shift per degree (~0.01%/°C), be very stable.

Thermocouples. A junction between two dissimilar conductors produces a small voltage difference, due to the difference in the effective concentration of electrons in the two conductors, which is almost linear with the junction temperature. Thermocouples are based on this property. They usually consist of two junctions, one is kept at a reference temperature (usually 0°C), the other is the temperature measuring sensor.
Thermocouples have a small size, a rapid response, a wide temperature range and are very simple. On the other hand they are not linear (although better than thermistors), the reference junction has to be kept at a constant temperature and they are self generating sensors, which means that their output energy is very low.

Silicon sensors. A forward biased diode also has a temperature dependent junction voltage. If the current through the diode is kept constant, this temperature dependence is exponential. Because the temperature characteristics for normal diodes are not highly reproducible, these diode sensors are not suitable to be used as high-accuracy thermometers. In applications with less demands, such as overheating detectors, they can serve very well.
A more advanced silicon sensor makes use of this temperature dependence of the junction voltage. The temperature response of the silicon diode is linearized by a signal processing circuit added on the same chip. The in this way accomplished linearization is in the order of 0.1°C in the temperature range -55 to +125°C.

Optical techniques. The thermal radiation of objects can be used to measure the temperature of these objects with infra-red sensors. This optical way of temperature measuring can be used in a very wide temperature range with a very high accuracy (e.g. one sensor can cover the range -25 to +1400°C with a accuracy of 0.2°C).

This non-contact method clearly has the advantage that it is non-invasive and even can be used at large distances. It led to the development of infra-red vision. With infra-red vision an image of an object can be produced, revealing the temperature distribution of that entire object.
Obviously, infra-red camera's give powerful abilities to industry, medical science etc.
The disadvantage of the infra-red method is the complexity of the signal processing circuit, especially for infra-red camera's. This makes the method less useful for control applications.

7.3.1.3. Flow

Existing flow measurement methods are based on measurement of: differential pressure, heat exchange, mechanical effects, Doppler effect or cross correlation.

Differential pressure. If the flow is measured with the differential pressure method, the pressure difference between two points along the stream, if necessary separated by a constriction or other aerodynamic mechanism, combined with the flow-through area and some physical properties of the fluid or gas (such as viscosity) is used to calculate the flow. Difficulties that can occur with this method are: calibrating problems and variations in viscosity due to variations in material density, composition and temperature.

Heat exchange. A thermistor with a deliberately high applied voltage or current and thus high internal power dissipation, is cooled by a flowing fluid. The cooling rate is proportional to the square root of the velocity of the fluid. By keeping the resistance (and thus the temperature) of the thermistor constant by means of a feedback circuit, the applied voltage or current can be used for determining the flow.
Usually, a second thermistor is needed for compensation of temperature changes of the fluid.
Several silicon sensors, with advanced compensating and signal processing circuits, based on this principle are developed. If a heating device is centralized in a circle of thermal sensors, not only the velocity but also the direction of the flow can be determined.
Disadvantages of this method are the need for a compensating thermistor, the method is sensitive for variations in material density, it is invasive and it is not possible to combine a fast response time, which is satisfied by a small fragile thermistor, with rigidity demands, which in turn will lead to solid thermistors which have a long response time.

Mechanical. The mechanical method is based on measuring the revolving speed of a turbine that is localized in the flow. This method is very invasive and problems such as sticking, choking-up, corrosion wear etc. can occur.
The advantage of this method is its simplicity.

Doppler effect. The Doppler flow-measuring method is based on the well known Doppler effect. This method can be used for ultra-sonic and electro-magnetic waves.
This modern advanced method has the advantage that it is non-invasive and has an order of magnitude less dependence of viscosity, density, temperature and composition.
The disadvantage of this method is the higher complexity compared to the above-mentioned methods.

Cross-correlation. The cross-correlation flow-meters (Beck, 1983), measure the transit time of a tagging signal in the flow between two separated sensors along the stream. Dependent of the sort of tagging signal, these can be optical, sonical, thermal or electrical

sensors. A tagging signal can be already available in the flow due to inhomogeneities or it has to be inserted (such as a thermal fluctuation).
The advantage of this method is that the flow of very inhomogeneous liquids or gasses can be measured. There is no need for calibrating the sensors since only the transit time is the relevant parameter. The optically detectable tagging signal is also non-invasive.
The disadvantage of this method is the higher complexity, it is even more complex than the Doppler method. Due to the reduction of the cost of VLSI-circuits and microprocessors, this method has become economically realistic.

It is a remarkable fact that the differential pressure method, in spite of its problems, still is the most widespread flow measuring method. The simple mechanical method is second in order of occurrence. The Doppler method is beginning to find its way in industry, but the most advanced cross-correlation method still seems to be in development.

7.3.1.4. Level

There are several methods to measure the level of liquids. This can be done in a *discrete* or a *continuous* way. In the first case, the sensor only indicates if the level of the liquid is above or below its position. In the second case the sensor gives an indication of the exact level of the liquid.
Among the wide variety of level sensors, one has to choose the proper sensor for a certain application dependent on the property of the liquid and its container.

Thermistors and piezo-electric crystals can be used for discrete level sensors. The *thermistor* as a discrete level sensor is based on the difference in cooling effect between liquid and air, which can be detected in the same way as the flow measuring method. The *piezo-electric crystal* as a part of an oscillator circuit can be used as a discrete level sensor by detecting the resonance shift of the frequency of the crystal or the lowering of the amplitude of the oscillation, caused by the surrounding liquid.

Continuous level measuring methods are: differential pressure, displacement of a float, resistance measurement, capacity measurement and optical techniques.

Differential pressure. Among these measurement methods, the differential pressure method, based on the pressure caused by the fluid, is the most widespread. This method has the same sort of disadvantages as its analogy in flow measurements, such as problems caused by calibration, variations in material density, composition and temperature.

Displacement of a float. The mechanical method, based on displacement of a float connected to a potentiometer or other sort of displacement transducer, also is a very common level sensing method. The disadvantages of this method are that it is an invasive method and problems such as corrosion, wear and sticking can occur. The advantage of this method and the reason why it is so frequently used is its simplicity.

Resistance measurement. The resistance measurement method is a simple method which can be used with conducting liquids. The resistance measured between a dip-stick and the liquid is inversely proportional with the level.
Variations in conductivity of the liquid can be compensated by adding a reference pair of electrodes in the liquid.

Capacity measurement. The capacity measurement method is a method that is based on the dielectrical properties of the fluid.

Varying properties of the liquid, can be compensated by making use of a reference capacitor in the liquid.
This method cannot be applied with conducting liquids. It is a rather simple method, but due to demands on the electrical properties of the liquid less frequently used than the above-mentioned methods.

Optical techniques. Level measurements with optical techniques have the advantage of being non invasive, but problems such as sticking and obscuration and (if the level method is not a simple "reach level" measurement) also calibration problems can occur. In spite of this the optical method seems to be attractive enough to explain its increasing application.

7.3.1.5. Composition

In the chemical industry a lot of effort is put in the development of special dedicated sensors used for measuring compositions of solutions (e.g. PH-measurement). Usually these sensors are based on measuring the voltage difference of two different electrodes, sometimes shielded from the solution by a membrane, permeable to specific ions to be measured.
This sort of sensing has aspects specific for chemical industry and is considered to be of less interest in this context.

7.3.2. Sensing in Mechanical Industry

In the mechanical industry sensors are used to measure parameters which are in general different compared to those measured in the chemical industry. These parameters typically are temperature, distance, presence, force, torque, friction, length, area, angle, displacement, surface roughness and texture.
The major tasks of sensors in mechanical industry nowadays are detection tasks, such as detection of abnormal conditions, detecting the presence of objects, detection of tool wear etc. The result of these tasks usually is a right/wrong message.
Sensors are also used for measuring tasks, in order to check on the work being undertaken. In this case the decision about right and wrong is taken at a higher level. This implies that more information has to be gathered than in the case of simple detection. This is the reason why this is a more complex task than simple detection.
Recently, even more complex tasks such as localizing and recognition of workpieces and quality control have become possible. It may be clear that these rather complex tasks require collecting of even more information.
All of these tasks are accomplished by means of measuring one or a combination of the above-mentioned parameters with sensors suited for the application.

So far, the mechanical industry has shown little interest in using sensors for the more complex tasks. Apparently the current attitude is: as long as limit switches can do the job there is no need to replace them for more complex systems.
Using more advanced sensors in a system can result in a more flexible system. For instance if the decision about right and wrong is taken at a higher level, this usually is a software level and thus easy and cheap reprogrammable.
No doubt the cost of a system with more advanced sensors initially will be higher than a system with simple sensors, but the increase of flexibility obtained can in the long term result in economical benefits.

Next a discussion is given about some methods used to measure some of the above-mentioned parameters.

7.3.2.1. Temperature

Temperature measurements in mechanical industry require, due to the different aspects of this industry compared to those of the chemical industry, a different approach. Nevertheless the basical principles of temperature measurement are the same for both industries and the same sort of sensors can be used, although probably with a different design. Sensors that can be used for temperature measurement are described in section 7.3.1.2.

7.3.2.2. Distance

Distance of objects can be measured with inductive and capacitive proximity detectors, ultra-sonic methods and optical methods.

Inductive proximity detectors. The inductive proximity method is restricted to objects with certain magnetic properties.
This method can be based on measuring the reluctance of a probe coil, that indicates the distance of ferromagnetic objects.
It can also operate by subjecting electrical conducting objects to an alternating magnetic field. This field generates circulating currents (eddy-currents) in these objects. These eddy-currents in turn create a magnetic field, that interacts with the subjected field, thereby affecting the impedance of the coil that generates the subjecting field. Measuring this change in impedance can result in an indication of the distance of the object.
This sort of distance measurement of ferromagnetic or electrical conducting objects is only applicable for rather short distances (up to 10 cm) and in a magnetic distortion-free environment.

Capacitive proximity detection. Capacitive proximity detection is based on measuring the capacitance formed by electrically conducting objects and a probe. This capacitance is inversely proportional with the distance between them.
It may be clear that this method puts heavy electrical constraints on the environment. For this reason it is not a widespread method.

Ultra-sonic method. The ultra-sonic method is based on measurement of the elapsed time between the sending and receiving of a (modulated) ultra-sonic wave. This ultra-sonic wave is send out by a source and is reflected by the object back to a detector. The elapse-time together with the knowledge of the traveling speed of the sonic waves, defines the distance of the object.
This ultra-sonic method has proven to be a reliable system for range-finding applications in the range 0.01-10 meter. The method hardly puts constraints on the environment and material of the object.
Although the method is commercially viable and already used in numerous industrial applications, quite a lot of effort is still put into research and development. The main development effort is put into the direction of making the method applicable for robots.
An interesting direction of research is the field of acoustic imaging. Primitive shape recognition is possible using a matrix of ultra-sonic transducers. It is also possible to do texture determination using ultra-sonic methods.

Optical Method. Optical methods to measure distances are based on the triangulation principle. A beam of light is cast on the object and the angle between the direction of the light-source and detector and the distance between them is used to calculate the distance of the object. Range finding based on this principle can also be one of the tasks of a complex general-purpose vision system.
A number of dedicated range finders are developed using this method. These range finders in most cases make use of laser beams, in order to reduce necessary constraints put on the environment.

7.3.2.3. Presence

The probably most widespread type of sensors in the mechanical industry are sensors used for presence/absence detection. These type of detectors are often very crude, in which case they merely consist of *contact switches* or *optical beam interruption switches.*
More elaborate systems can distinguish metallic and non-metallic objects. This can be done with simple *metal contact detectors,* making use of the electrical conducting properties of metals or with inductive and capacitive proximity switches, making use of the electrical and/or magnetical properties of metals like distance sensing.
It is surprising to see how much is achieved using this simplest form of sensing.

7.3.2.4. Displacement

The change of position of an object is an important parameter, which can be measured with displacement transducers. There exists a wide variety of these transducers, based on different principles. They can be subdivided in linear, which sense the displacement along a line, and angular displacement transducers, which sense rotation about an axis.
A very common application for this sort of transducers, is to be combined with some flexible element, that changes its position under influence of some external parameter. In that case, not the displacement itself, but the external parameter that causes the displacement is the subject of interest. Examples of this are discussed in 7.3.1.1 and 7.3.1.4.
There are some displacement transducers, which do not require a mechanical contact with the object, but most of them do. In that case one has to be careful to select a transducer that does not noticeably influence the measurement due to friction or elastical forces.
Some of the displacement transducers are: potentiometers, inductive and capacitive displacement transducers, strain gages and digital displacement transducers.

Potentiometer. A potentiometer is one of the simplest displacement transducer. A constant voltage is applied across the potentiometer and it delivers an output voltage proportional to the displacement. Obviously, due to its construction, it can easily be used as an angular displacement transducer. By means of a cable system however, a potentiometer can quite simple be transformed in a linear displacement transducer, with a range in the order of 10cm.
Disadvantages of potentiometers are friction and a rather poor resolution ($\sim$0.1%). The advantage of using a potentiometer as a displacement transducer is that a simple readout circuit suffices and the method is linear in its full range.

Inductive. Inductive displacement transducers are, like the inductive proximity detectors (7.3.2.2) based on the change of inductance of a coil. In this case however, the

ferromagnetic material that changes the inductance by changing its position is part of the transducer itself. The object, of which the displacement has to be measured, is mechanically in contact with this ferromagnetic material. Due to this, there are no requirements put on the material of which the object is made.
Inductive displacement transducers usually are linear displacement transducers. These transducers need a more complex read-out circuit than the potentiometer. They have a limited range in which the transducer has a linear behaviour.

Capacitive. Just as the inductive displacement transducers show a resemblance with inductive proximity transducers, capacitive dislacement transducers show a resemblance with capacitive proximity detectors (7.3.2.2.). In this case the moving electrode that changes the capacitance is also a part of the transducer and is mechanically connected with the object. These displacement transducers can, dependent on their construction, be used for angular or linear displacement measurement.
An advantage is their high sensitivity, disadvantages are the rather complex read-out circuit required and non-linearity.

Strain Gages. If an object is subjected to stress, this object is deformed. This deformation can result in a change of the length of a part of its surface. This change of length can be measured with a strain gage, cemented on that surface.
A strain gage is an electrical resistor, which changes due to strain. This is the result of the change of cross-section of the conducting strip which forms the resistance. If the gage is made of semi-conducting material, the strain influences the resistivity of the material itself, usually resulting in a higher sensitivity.
Strain gages are very sensitive and do not require a complex read-out circuit. On the other hand, they are also very sensitive to temperature, which may cause requirement of compensating measures (Wobschall, 1979).

Digital. No doubt a transducer, that provides a digital output signal, is desirable in this world of increasing use of digital systems. There are both angular and linear displacement transducers that provide such a digital output signal. The angular digital encoder is a displacement transducer. It consists of a disk, which has a number of tracks on it, and a read-out system. Each track contains a bit of a digital coded signal, which represents the absolute angular position. This coded signal can be binary or some sort of cyclic code and is usually read-out optically. Joint position sensors as used in robots, usually are of this type.
The linear encoder is based on the same principle.
The resolution of both types of transducers is determined by the number of bits (and thus tracks) used (this may be up to 14). Using an additional track which produces a sine wave output, can increase the resolution with 5 bits (Woolvet, 1983).
Another digital transducer, based on the same principle, but with one single track, is the incremental type. In this type the number of pulses that the disk produces as it turns round are counted. In this way, the number of revolutions or the relative angular displacement can be detected. The resolution of this type can be very high. Linear incremental displacement transducers are similar.

7.3.2.5. Computer Vision

There are a number of dedicated computer-vision systems commercially available. They operate successfully on low level problems of verification, inspection, recognition and determination of object location and/or orientation. Low level problems are problems that are based on one or two dimensional images of objects in less structured environments.

A very attractive application of computer-vision in the mechanical industry is non-contact inspection of objects. It has the following advantages compared to the more commonly used contact inspection methods.

- It can be faster because the surface or contour of the object does not have to be scanned mechanically.
- The object cannot be damaged by the optical method.
- There is no wear of the probe, which would require recalibration.
- The object usually does not have to be positioned.
- Non-contact inspection can be a safe method, especially in the case of hazardous objects.

Even the most sophisticated computer vision system cannot handle problems such as recognition in complex structured environments in real time. Another restriction is that they are not equipped to handle three-dimensional analysis for recognising objects from arbitrary viewpoints.
There is no general purpose vision system commercially available today.
More on this subject in section 7.6.

7.3.3. Robot Sensing

The first generation of robots (around 1965) were used in limited area's of industry. This was mainly due to the fact that they were equipped with internal sensors and the only interaction with their environment was for synchronization. Internal sensors usually are rather simple sensors such as joint position sensors. This usually results in rather simple and thus inexpensive control systems. On the other hand, using internal sensors requires a high positioning accuracy of the objects to be handled, which results in costly object fixing systems.
Current robots (Gevarter, 1982) may have interface provisions for external sensors, to be used for feedback control. The feedback facility of these second generation robots enables them to respond to changing operating conditions, which allows expansion of their task-range. External sensor systems can be rather complex, such as force, proximity or vision systems. Clearly, using these sensor systems will imply higher cost due to the real-time demands on the processing of the sometimes rather complex signals they provide. On the other hand, these external sensors yield a much higher flexibility, which can be the reason that using these external sensor systems may turn out to be economically beneficial.
This is the first step towards robots of the third generation. These are robots which are able to determine their own actions based on their perception and planning abilities.

When it comes to robot sensing quite often the distinction between contact and non-contact sensing is made as stated in 7.1. Non-contact sensing provides important information for manipulation control. The most powerful non-contact sensor system is vision. Some specific information needed for manipulation however, cannot be obtained with vision systems at all, or only with extreme effort (and cost). This is typically the kind of information that can be provided by means of contact or near-contact sensing.

7.3.3.1. Contact Sensing

The simplest contact sensors used for robots are switches that stop arm motion and open and close grippers. More sophisticated contact sensors can measure slip, force, torque, displacement, temperature and surface roughness. They can even be used to recognize objects.
Contact sensing for industrial robots is still at a rather primitive level. If force is measured at all, it is measured with simple mechanical sense-probes based on measuring displacement caused by stress, or by measuring internal signals such as air pressure in pneumatically operated gripper systems. It is surprising that contact sensing not always takes place in that part of the robot, where the robot mechanically makes contact with the outside world, that is its gripper fingers. This is especially the case for the less sophisticated force sensing systems. They usually are build in the wrist of the robot and merely indicate whether the gripper is touching an object. Usually they can tell in what direction, with what force the gripper is pressing against this object.

In various research laboratories however, rather sophisticated touch sensing systems are being developed (Stânton, 1983). Most of these touch sensors are based on measuring resistance variations between array's of electrodes, induced by deflection of for instance a pad of conducting foam (Christ, 1982). Other recently developed touch sensors are based on arrays of sensors made of semi-conducting material.
A big advantage of these sensors is that they can be mounted on the gripper-fingers. With these systems it is possible to detect shape, texture, slipping of objects lifted by the robot and compliance of objects by touch alone.

Perhaps one of the most vital problems for robots nowadays is the problem of soft grasping of objects. If grippers are made compliant to insure that objects are not damaged, this unfortunately results in undesirable effects such as a reduced frequency response and hysteresis of the force sensor and uncertainty about the position of the sensed object. If the compliance is imitated by the software to eliminate these effects, this requires costly high bandwidth sensors, hardware and software.
In industry this problem is evaded by using grippers that are dedicated to the particular application. It might be clear, that this need for dedicated grippers is a limiting factor for the universal applicability of robots for manipulation.

7.3.3.2. Non-Contact Sensing

Beside robot vision, there are a number of non-contact sensing methods, that have shown to be very useful for robot applications. Most of these non-contact sensing methods however, are restricted to short distance sensing. For this reason they are often referred to as near-contact sensors or even as remote touch. Examples of how to accomplish this near-contact sensing is described in 7.3.2.2.
This sort of sensing is specially useful for avoiding obstacles and for positioning the gripper prior to grasping.

7.4. Future Trends

There is no doubt that there is still a lot to be done to improve all types of sensors. Higher reliability, higher durability, non-invasiveness and a low cost are some of the properties, the industry thinks most desirable. A higher speed, a better resolution and a larger dynamic range are the main properties desired by the development laboratories. These are demands on the sensor hardware itself.
Even more desirable however, is improving the sophistication of the sensor signal processing.

For CIM, it is important that modular systems with standardized interfaces between them will be developed. This would make integrated systems of a variety of robots, tools, sensors and control systems possible. For sensor systems this would mean the standardization of sensor interfaces. Clearly due to the large difference of complexity of sensor systems, it does not seem advisable to design one standard interface covering the whole area.
It might be better first to design standards for a simple sensor interface, which is commonly thought to be ready for standardization. The next step would than be the design of a complex sensor interface standard, which probably would be too early to do at present.

In laboratory environment, tactile sensor systems already have proven to be able to offer the necessary versatility and flexibility for robot grippers. This will allow robots to be used for more complex assembly applications.
One can expect that due to the amount of research in progress, it probably will not be long before tactile sensor systems are ready to be used in industry.

7.5. Processing for Sensors

Sensor processors can be associated with the various sensing levels indicated in fig. 7.1

The individual sensors as a rule have a built-in (simple) processor for filtering and tuning the raw signal.

The next higher level typically is a real-time data acquisition type processing, which can involve anything from simple to really intelligent processing. Important for further processing is to only collect meaningful signals.

When, on a higher level, coordination of multiple sensors for a complete sensor system is required, than this, in the case of detection, is done preferably in the same local microprocessor. Also in this case interpretation for the purpose of filtering out uninteresting signal combinations as well as (periodically) resetting and sampling the data is controlled by these local processors.

The more intelligent ones can accept input from higher level control, for instance, for specification of interesting sensor patterns.

Every sensor system must be able to be connected either in a polling mode (i.e., sense once on request), a sampling mode (i.e., sample continuously or at a given frequency) or in an event mode (i.e., signal an alarm when a signal is sensed). This provides sufficient flexibility to use such a sensor system in arbitrary control systems.

On the next higher level the sensor application can be divided in data and control. The processing characteristics on this level usually are dominated by the overall application to which the sensor contributes.

Detection type sensors in many cases are used for simplifying tasks and the corresponding processing. For, instance a proximity sensor can be used for very accurate positioning. This is most of the time much easier than trying to very precisely calculate the absolute control signals for the servo system of the moving device. This implies that calculations for such detection type sensing is relatively simple.

Vision systems which are applied in this manner, often referred to as vision-detection, are assumed to be of different type than the so-called pure vision systems which always perform some kind of scene analysis. They will be discussed in the next section.

7.6. CIM and ROBOTIC-VISION

This section consists of two parts, a general Introduction to CIM and Robotic-Vision. The Introduction provides background material which may be omitted if only the Vision part is deemed necessary.

7.6.1. Introduction

In this section the role of Robotic-Vision in CIM will be considered. To establish a frame of reference and to provide a provisional definition of various terms we shall first review CIM briefly. At the highest level, Industrial Manufacturing (MFG) is a black box that converts a raw material input to product(s). At a lower level, MFG is seen to contain several, more or less autonomous, modules e.g. Design & Engineering, Machining, Assembly, Transport etc. The role played by manual workers in different modules varies considerably. The aim of CIM is to minimize, if not eliminate altogether, the human role through IT (Information Technology) and achieve a degree of automation, With the current IT, it is possible to expect a very high, if not total (in some cases), automation in production i.e. in Machining, Assembly etc. Design & Engineering of products has to be inherently flexible and thus has a high level of human involvement. Automation in this field i.e. Robots designing products, seems far-off. In this sense, Flexible Design & Engineering is truly Computer Aided Design & Engineering CADE, rather than Automated. This causes a sharp division between CADE and Computer Aided MFG (CAM) as shown below.

C I M	
CADE	CAM
Interactive Software	Turn-Key
Menus	Servo Control
Graphic Displays	Sensors
Simulators	Image Processors
Solid Modellers	Scene Analysis

Figure 7.2
CADE and CAM

IT needed for CADE has a flavour very different from the one for CAM. For CADE one needs Interactive Software with MENU's joysticks, lightpens, graphic displays and so on. The computational emphasis is on numerical modelling of shapes and figures, simulation of physical conditions. For Flexible design of products, human intervention is an indispensable ingredient of CADE. For CAM, Turn-Key Software, with minimum human intervention, is the rule. In Fig. 1, the sensing layer is also put in the proper CIM perspective. It is comparable to the display layer of the design phase. To allow flexibility in manufacturing i.e. to keep the costs down as the product line changes; one introduces the concept of a Flexible Manufacturing System (FMS) which can be defined as (Reddy, 1983)

- FMS is a programmable batch-processing arrangement that contains programmable machine tools and transfer devices, all under the control of a central computer.

In practice, to avoid sympathetic failure of the entire production line due to a single component failure, i. e. to make FMS fail-soft, the system consists of Task Automation modules or FM-Cells (FMC). An FMC is an Autonomous Automated Production Unit built around ROBOTS. An internationally accepted definition of a Robot is (Flinchbaugh, 1983)

- "A Robot is a reprogrammable multifunctional manipulator designed to move material, parts, tools or specialized devices, through variable programmed motions for the performance of a variety of tasks".

From a practical point of view one tries to replace manual labour by Robots to achieve an increase in the productivity. It is well known that about 70% of the manufacturing effort (measured in dollars and time) is devoted to material handling e.g. loading-unloading, inventoring, material flow control, workpiece/tool transport etc. These tasks are typified by such problems as pick & place, bin-picking, visual inspection etc;, and make liberal use of human skills in hand-eye coordination, visual perception and decision making. To develop an Automated Transport system, Sensors of different type are expected to play a vital role. For these tasks, the Robotic features acquire a strong 'Anthropomorphic' flavour, as can be seen from the following list of the typical Robotic features;

- End Effector: Gripper, Fingers, Vise, Suction Cup, Magnet
- Manipulator Arm: Shoulder, Elbow, Wrist (with yaw, pitch and roll), Joint Actuators, Booms
- Locomotion: Fixed, Wheels, Legs, Tracks, Girders
- Sensors: Tactile, Proximity, Environment, Vision
- Power: Pneumatic, Hydraulic, Electric.
- Control: Open Loop, Closed Loop Servos.
- Operation: End Stops, Programmed Lead Through, Walk Through, Software Addressed, Smart Sensor (Adaptive Control)

There is no standard taxonomy of Robots and different countries use different schemes to classify Robots. In Japan, the classification is based on the input and learning mode as shown below (Gevarter, 1982).

Teleoperator: --Copies remote operator movement and could hardly be called a Robot.

Fixed Sequence Robot: --Follows a fixed sequence of steps generally defined by physical stops and needs mechanical re-setting for changing the sequence and thus entails considerable down-time as also the attendance of a skilled operator.

Variable Sequence Robot: --Sequence changes can be implemented by re-setting dials, levers, tapes (NC-Machines) etc. Down-time for sequence change is much less than the above.

Playback Robot: --Can repeat operator movements. Has resident hardware and software to calculate and remember positions, conditions etc; for the operator movements in the teach-in mode.

NC Robot: --Can execute movement and position instructions coded in a relatively high level language.

Intelligent or Smart Robot: --Has Sensory (hardware) and Cognitive (software) capabilities to exercise adaptive control i.e. modify position and movement sequence as the conditions change.

Obviously the last category, Smart Robot, is the most desirable one and commands the highest share of R & D activities. In the US, a similar sort of classification is arrived at by first dividing the Robots into two major types;

Non-Servo Robot: --It is the traditional Pick & Place Robot with very few degrees of freedom. In its simplest form it is used for handling of palletized parts. All movements are controlled by preset electro-mechanical switches.

Servo-Robot: --Has servo-feedback loop to alter, in mid-air, the course of end-effector, arm etc. To effect these alterations one obviously needs the entire infra-structure of sensors and processors.

In Europe, there is no coherent approach to this problem and classification or identification of generic types of Robots is done in an ad hoc manner. Whatever classification is used, the status of individual components e.g. end effectors, arms, controls etc, can still be reviewed to assess the potentials and problems.

End Effector: --These are generally dedicated to the problem at hand and thus hundreds of different type are available. The two important factors here are; dexterity and signal. On the mechanical side, multiple fingers, articulated fingers etc; are actively pursued. For acquiring and analysing tactile signals artificial-skin, tactile sensor arrays etc; form a major R & D effort.

Manipulator Arm: --The basic goal of an Arm is to provide a Point To Point (PTP) motion by following a trajectory as given by, software, contours of a surface or determined adaptively. There are four basic configurations, Cartesian, Polar, Cylindrical and Articulated. These provide the usual six degrees of freedom, however, additional areas. In general, for a given PTP, there is a large number of trajectories possible. Finding the 'best' trajectory may require a large computational effort. To calculate and find the optimum trajectory in real-time is one of the important areas of R & D.

Locomotion: --Currently, locomotion seems to be at the expense of other attributes i.e. mobile robots are not dexterous or smart enough (just glorified coffee-carts), whereas dexterous, smart robots are fixed. Giving a smart robot the capability of 'Walk-Alone' seems to place excessive demands on compacting the power plant and the on-board computer. Although this is, at least in Japan, the most desirable new feature sought in existing Robots, not much advance is expected unless the dexterity and locomotion are individually well advanced.

Power Plant or Actuators. --To power the various limbs of a Robot three basic mechanisms are used; Pneumatic (compressible fluids e.g. air), Hydraulic (non-compressible fluids, oil) and electromechanical (electric motors). Pneumatics is simple and cheap but gives a 'squishy' effect and thus dimensional accuracy of the end points is not high or elaborate closed loop servo-controls are needed. Hydraulic drive is the most popular, but is messy and cumbersome. It requires holding and ballast tanks, bleed lines, finicky regulators and assorted plumbing fixtures that may require periodic maintenance and thus reduce the up-time. Electric drive is the easiest and cleanest but could not deliver enough power for many jobs. Major R & D in Mechanical Engineering is afoot to address this problem.

Sensors: --This is perhaps the most important aspect of Robotics today and if implemented successfully, it promises to usher in a new era in Automation. Whereas most of the mechanical anthropomorphic features, e.g. fingers, arms, legs; have been incorporated, to some operational degree, in current Robots, attempts to mimick human senses, e.g. vision, touch, hearing; have been successful only at the most rudimentary level. The hardware part of Sensors is in a relatively advanced state. One can get off-the-shelf Sensors to meet common requirements of sensitivity, resolution etc; and new types e.g. artificial-skin, are being developed at a fast pace (Sanderson, 1983),

The problem of Sensor Signal processing i.e. a vast amount of sensory signal to be processed in real-time, is the most nettlesome. To overcome this, major R & D effort on several fronts is underway. This activity can roughly be divided as follows.

Software Development: --The Robotic Software can be divided into three broad classes, namely, supervisory or guiding software, adaptive software and global or world modelling software (Lozano-Perez, 1983),

The first one is primarily for NC-Robots and calculates trajectory and plans motion sequence from manual input. This sequence is repeated by the Robot, irrespective of the 'world-situation'. The adaptive (or robot-level) software is primarily meant for sensor-based Robots. Its input is from actuator-transducers and sensors e.g. vision and force; which is processed to plan the trajectory to achieve a software-defined goal. This type of software is dedicated to the particular Robot-Sensor system being used and models the task in general. The highest level of Software, i.e. 'World Modelling', is independent of Sensor or the Robotic-Vision system used. Here, high level languages accept mission-commands e.g. put a pin in the hole of the square workpiece. In this case an appropriate Sensor (e.g. a vidicon), interprets a 'world-scene' and using a knowledge base identifies the needed objects, i.e. the pin and the square workpiece with the hole and calculates all the movement sequence and executes the command. At present, such level of performance is possible in advanced laboratory environment.

Hardware Development: --Signal processing for Robotic sensors is much too complex for a computer, small enough and fast enough to be attractive for Robotics. This work, however, is well-suited for parallel computers. There is a serious effort in this direction to develop compact parallel computing hardware. This activity will play a major role in the further development of Robots.

Smart Sensor Development: --The signal processing burden can be significantly lightened by the proper choice of the sensor. In general, a video sensor gathers a large amount of data that can seriously load the computing facilities. On the other hand after an initial step of low-level data processing a considerable data compression can be achieved. This 'preprocessing' can be be done at the sensor by either using physically-smart sensors e.g. optical filters or correlators, or building the logic on the sensor (Forchheimer, 1983),

A single chip linear array picture processor. SPIE 397(1983)000 A similar effect can also be realised by reconfiguring the sensor-source system. Using an ordinary vidicon in a passive mode, to 'understand' a 2D-image of occluded 3D-solids, will require elaborate signal processing. However, using the same sensor in conjunction with a Laser-Scanner or

Laser-Triangulator, would reduce the computational load considerably. The current trend in Sensor choice is to go for such solutions. A large variety of sensors are now available to meet different physical needs. Some commonly used sensors are listed below.

Optical: --Fiber-Optic relays, LED Proximity Sensors, 3D-Stereo optics, Colour Vision, Fourier Correlators, Matched Spatial Filters, Laser & Holographic Triangulators, Speckle Texture Analysers.

Thermal: --IR-Vision, Focal Arrays, LCD-Thermo- Chromography.

Sonic: --Ultrasonic Ranging, Echography, Acoustic Tomography, Acoustic Holography, Phased Array Scanning, Acoustic Microscopy, A, B, C-Scans, Photo-Thermo-Acoustic Diagnostics.

Tactile: --Contact Sensors, Piezoelectrics, and Piezoresistivity, Presure Transducers (Presure-Transistors), Si-Array, Mechano-Electric Transducer, Strain-gages, Resistive Foam, Artificial Skin.

Proximity: --Capacitance, Magnetic, RF-Inductance.

Of all the sensors, visual sensors, keeping in with the important role played by the human vision, are the most desirable sensors in Robotics. Various disciplines, such as Robotic Vision, Computer Vision, Machine Perception; have emerged to address this problem. A brief review of this is offered in the following.

7.6.2. ROBOTIC-VISION

A rigorous definition of Computer Vision does not exist, except as a statement of its goal in anthropomorphic form, namely, Computer Vision is the art and science of developing systems to imitate human visual perception. For Robotic-Vision (a sub-set of Computer Vision) this can be further narrowed to the definition of a black box whose input is a 'scene' and the output is a set of labels assigned to sub-Images of the scene. Generally these labels belong to a knowledge base (or, a pool of a' priori knowledge) supplied by the user. These labels can have varying degrees of depth e.g. as a simple label 'an object', or as a deeper label 'a pump housing with four mounting holes lying on the conveyor belt with its axis 57° off the belt axis'. This could then help in performing such common industrial tasks as;

- Recognition of workpiece and/or tool,
- Determination of their pose,
- Salient feature extraction for spatial reference,
- In-process inspection, quality control.

To accomplish these tasks, Robotic-Vision can be partitioned into two distinct modules;

- Image Acquisition, and
- Image Understanding.

In Image Acquisition a 3D-scene is projected onto a Photo-Electronic device as a 2D-photon distribution which is converted into electrical signals, usually in the form of a matrix of digitized grey values. In Image Understanding this matrix is mathematically processed, often in conjunction with a resident knowledge base, to arrive at the labels. Historically, Image Acquisition has been the domain of Photo-Electronic Hardware, whereas, Software (and the associated computer Hardware) reigned in Image Understanding. As a result, these two fields developed pretty much independent of each other. After a brief review of these fields, we shall attempt to show that the recent activity in Smart-Sensor is merging these two i.e. there are Photo-Electronic devices that can produce, from a scene, not only a matrix but also labels, albeit shallow ones. This is a major development that promises to play a significant role in Robotic-Vision.

Image Acquisition

In Robotic-Vision, Image Acquisition by definition, is a real-time activity and thus the traditional means of Imaging e.g. photography, are not considered. There is a large number of devices available for Image Acquisition. They differ widely in their characteristics and can meet diverse demands. There is no unique classification scheme to group them into well-defined categories. We shall use device physics and the following general properties to classify them.

- Spectral Sensitivity; is the ability to respond to radiations of different wavelengths and can be measured as the signal output per input photon of a given wavelenght. The wavelenght can vary from the macroscopic i.e. about a meter for SLR (Side Looking Radar), to 1.0E-17m for nuclear radiations. If Robotic-Vision applications are envisaged from Space Explorations of the Earth to nuclear plant operation, then this entire range has to be kept in mind. In intensity, the incident photon-flux could range from single photons (photon-counting) to solar levels (1.0E+18 photons).

Obviously no single device has the dynamic range to deliver such performance.

- Resolution; can be taken as the number of spatially resolved elements (pixels) into which a scene can be divided by the Sensor. A high grade, off-the shelf TV-Sensors can now deliver up to one million pixels per scene (photographic film gives about 1000 times more pixels) (Blouke, 1983), This is enough for almost all current applications of Robotic-Vision, where the bottleneck is that there is not enough computer power to digest so many pixels in real-time. In current Robotic-Vision practice, an Image of more than 64k pixels is seldom considered.

- Speed; is the Sensor's ability to follow scene motion. Since, the present-day robots are still very anthropomorphic, a fraction of a second can be taken as a benchmark time interval. An ordinary TV-Sensor with a frame time of 25 ms is fast enough for Robotic-Vision. Again, owing to the limited real-time computer capacity available presently, a quantum jump in the speed is not expected and the Sensor speed is not likely to be taxed soon.

- Signal Mode; this is a catch-all phrase to describe how and in what form is the video signal generated. A TV-Sensor gives its signal in a sequential (raster) mode, however, with some modifications, it can address the scene as a Random Access Video Memory. With further modifications the video signal can be read with 'Logic', i.e. some low level Image processing can be done at the Sensor. Such developments have strong implications for Robotic-Vision and one should be alert to advances in device-physics; which will be briefly discussed next.

Device Physics.

Based on the physical principles of operation, the Robotic-Vision Sensors can be divided into two broad categories,

- Electron-Beam addressed Sensors, and
- Charge Transfer Devices (CTD).

Electron-Beam Sensors.

Electron-Beam Sensors are the old war horses of the vacuum tube era (Lubszynsky, 1978). Inspite of the latters demise, Electron-Beam Sensors are still unchallenged in many applications. The common principle of operation shared by these devices is that the incident light, from the scene, is converted into photoelectrons and onto an integrating solid state target. This charge distribution is 'read' out as an analog signal by an Electron-Beam that scans the target. In this sense these devices can be likened to an Electron-Beam addressed analog video memory. Whereas the 'read' mode for these devices is always the same i.e. Electron-Beam, there is a great diversity in the 'write' mode. From this stems the unchallenged versatility of the Electron-Beam Sensors. Using the 'write' mode as a distinguishing feature, the Electron- Beam Sensors can be placed into two large classes as discussed below.

- Vidicons. In these devices the light is incident on one side of a thin target and converted into a charge distribution. An Electron-Beam scans the target from the opposite side and 'reads' the charge distribution (in a destructive manner) to produce the analog video signal. These are all-round robust devices of simple construction and with good characteristics for general purpose applications. These include such devices as Antimony-vidicons, Plumbicons, Saticons, Newvicons etc (Goto, 1974). With moderate light sensitivity, they offer the highest resolution. For Robotic-Vision it is the first choice as an economic and good performance Sensor.

- Orthicons. This is a large family of rather complex, but highly specialized Sensors. The 'write' needs, generally, three components;

  1. A photoelectric target that converts the incident light into free electrons.
  2. A relay of electron-optics that focusses, accelerates, pans, zooms etc; the photoelectron beam.
  3. A solid state target integrates (or differentiates) the incident photoelectron beam into a charge distribution which is stored until read out by the Electron-Beam.

By selecting various combinations of these steps a wide variety of operating characteristics can be achieved to meet the most critical applications. Notable examples of this family are, SIT (Silicon Intensified Target), Image Dissectors, FLIR (Forward Looking Infra Red), FOC (Faint Object Camera of Space Lab), SEC (Secondary Electron Conduction, Apollo Moon Lander and Space Robot), ICCD (Intensified Charge Coupled Device). For Robotic-Vision the most noteworthy feature is that in spite of (or because of) its complexity, it offers great possibilities for on-site logical operations at the Sensor.

Charge Transfer Devices (CTD)

The development of a self-scanned solid-state Image Sensor, to equal the performance of an Electron-Beam Sensor has been an elusive goal for the last 20 years. Even the most advanced Chip-Technology has not been able to produce a cost effective Sensor whose picture quality is comparable to a modest Electron-Beam Sensor (Weimer, 1983), Although the sensitivity of a CTD is superior, its resolution does not reach the level of Electron-Beam Sensors. CTD's have, thus, not found wide applications in CCTV (Closed Circuit TV) or broadcast TV. However, a large number of Robotic-Vision applications do not place such exacting demands on resolution and can thus benefit from the unique features of CTD's which will be briefly discussed below.

A solid-state CTD Sensor, along with the photon conversion and charge storage functions of an Electron-Beam Sensor, has to also have a mechanism for scanning the charge distribution. To accomplish this, the (quasi)continuous Si wafer is graticulated with composite elements that act as both photoelectric converters as also charge storage elements. In one form or the other these elements are reverse-biased photo-diodes. These individual elements, the potential pixels, are arranged in either a lattice (to furnish a 2D-detector) or a linear array (facsimile scanners, linear-scanners). The incident photon flux from the scene is sampled at these sites and converted into photoelectrons. These photoelectrons are trapped in the diodes. There are two basically different modes to handle these photoelectrons, either by digital multiplexing or by charge transfer. Correspondingly there are two generic types of CTD's, namely, CCD (charge coupled device) and CID (charge injection device). A brief discussion of these follows.

- Charge Coupled Devices (CCD)

In these devices along with the diode-complex, a buffer memory in the form of a one-line memory buffer register, or a whole frame buffer memory is provided. The charge stored in the reverse-biased photodiodes is transferred into the buffer memory which in turn is 'read' separately.

- Charge Injection Devices (CID)

In these devices there is no buffer memory on the chip but instead there is an X-Y address bus. The reverse-biased photodiodes act as memory cells and can integrate low-level light. For all practical purposes a CID can be considered as a Random Access Analog Video Memory.

For all such Si-based Sensors the resolution is limited by the number of diodes. Presently there is an economic limit of 64k pixels, though chips with ~ 0.5 Mpixels, (800x800) is available for critical applications (NASA, DOD). A further limitation is that the spectral response is limited to that of the Si-substrate which may be too limiting for some applications. On the other hand efforts are underway to make optical elements (diodes) form a part of a dedicated Image processing chip. In this manner one can calculate Fourier Transforms or Correlations at unprecedented rates. In the context of Robotic-Vision the Sensor Hardware is in an advanced stage of development but much effort is needed to find the appropriate Software for Image Understanding and then integrate it into the Sensor chip. A brief discussion of the Image Understanding now follows.

Image Understanding

In human beings one follows a bottom-up approach to Image Understanding. We have, at the lowest level the physical perception of the Image by the retina (Sensor) and at the highest level cognition of the Image. The Robotic- Vision activity can also be organized into loosely ordered range of representations of higher and lower levels,

One can categorise the representations into four levels as follows.

- Iconics,
- Segmentation,
- Geometry,
- Relations.

A brief discussion of these will now be given.

-Iconoics. In this step the Sensor signal is processed to produce Iconic (Image-like) structures. This is the first step to which the Sensor data is subjected. Since different types of Sensors may be used, this step is Sensor-sensitive, e.g. laser-triangulation data has to be treated differently from the CCD data. In general, these are domain independent processes such as smoothing, contrast stretching and compression, histogram modification, entropy maximisation, thresholding, erosion, dilation, skeletonization, chain coding, various types of gradient enhancements etc. This is a very numeric-intensive step and the number of steps is proportional to an exponent (commonly ranging between 2 to 4) of the Image size i.e. the number of pixels. Obviously, the Sensor resolution is important here. Often a higher Sensor resolution is desirable and physically possible to deliver, however, it may require prohibitive amount of number-crunching steps and is hence not advisable. On the other hand, it is possible to carry out these operations quite efficiently if parallel architecture is used. There is thus a major R & D effort to develop dedicated parallel machines to carry out complex Iconic tasks. On the whole, whatever else is developed in Robotic-Vision for higher level tasks, if no efficient solution for this step is found, all the other efforts will be largely futile.

-Segmentation. In this step, using some a'-priori knowledge, some interesting features of the Image are sought and enhanced. This includes such Image processing operations as, edge detection, curve (and straight line) detection by Hough Transform, Graph search, contour following, region growing, split and merge, texture analysis etc. The result of this step is to replace the original Image-matrix by a set of geometrical primitives such as, lines, curves, boundaries. This leads to a very significant data compression. and can considerably ease the subsequent analysis. The operations required here are again number- crunching types combined with a bit of knowledge base. Many of the comments, offered for Iconics, are also valid here, in particular, a non-Von Neumann, parallel machine is quite desirable.

-Geometry. After the interesting features i.e. the geometrical primitives, have been isolated, in segmentation, the next step is to combine these primitives into geometrically identifiable surfaces and objects. This calls for 3D-reconstruction, boundary representations, polyhedra occlusion, strip trees, region representation, run lenght coding, spline-surfaces, shadows, structured light, computer generated Moire', line drawings etc. There is a very strong overlap with computer graphics at this stage. Much attention is devoted to this step in such Robotic- Vision problems as, bin-picking, pose and general assembly tasks. In a slightly different manner this step is also very important for Robotic-Vision-Quality-Control. Using special Sensors and/or special Software, features relevant to quality control are enhanced and isolated and then the quality of the product can be assessed. Commonly this task is not considered to be a part of Robotic-Vision but relegated to in-process quality- control. However, with some modifications this could be integrated into Robot operations, e.g. a painting Robot that performs quality-control of the finished surface while painting it.

-Relations. Once the interesting objects have been isolated and reconstructed, one tries to 'understand' the scene. This is generally done with the help of a knowledge base. This is, at the moment, the most difficult task. There is hardly a commercial Robot which can 'understand' a nontrivial scene. The operations used here are the domain of AI and include such items as, analogic and prepositional representations, semantic nets and inference matching, graph-theoretic matching, backtrack, decision tree etc. In general these operations seem to posses an inherent serial control hierarchy and need large knowledge bases. Current industrial Robotic-Vision practice does not yet extend much this far. The trend there is to come up to the Geometrical part and then devise some other configurations e.g. strip lighting, stereo-cameras, and engineered environment etc; to avoid this AI bottleneck.

Commercial Robotic-Vision systems, available today, are almost entirely based on binary Image analysis of silhouetted objects (engineered environment). The principle mode of operation is to start with a 'shallow' grey Image (2-4 bits) and threshold it to obtain binary silhouettes of the objects. For this binary Image, simple geometrical properties e.g. perimeter, area, curvature, low order moments etc; are calculated and a look-up-table type of feature-match is made.

7.7. Processing for vision

The processing required for vision is, as was already indicated in the previous section considerable. However, the expected pay off justifies development of special purpose hardware for vision. Moreover, it also justifies higher level number crunching type processing for the purpose of scene analysis, combined with data bases for recognition support.

Ultimately, a fully automated factory with built-in high flexibility will apply many kinds of vision systems, probably on more than one level of application, for instance for parts handling but also for monitoring.

It is at this stage not very well possible to give a much more detailed strategy for vision processing as almost all vision systems for CIM are still in the experimental stage.

However, research and development of such systems is absolutely necessary because of the enormous influence successful vision systems will have on the competitiveness of CIM subsystems.

References

Beck, 1983. Beck, M.S., "Correlation in Instruments: Cross Correlation Flow Meters," in *Instruments Science and Technology*, ed. B.E. Jones, Adam Hilger Ltd, Bristol (1983).

Blouke, 1983. Blouke, M.M., *Opt. Engg* **22** p. 607 (1983).

Christ, 1982. Christ, J.P. and Sanderson, A., "A prototype Tactile Sensor Array," The Robotics Institute TR 82/14, Carnegie Mellon University, Pittsburg (1982).

Flinchbaugh, 1983. Flinchbaugh, D.E., *IEEE NAECON Conf.* **Part 1** pp. 184-188 (1983).

Forchheimer, 1983. Forchheimer, R. and Odmark, A., *SPIE* **397**(1983).

Gevarter, 1982. Gevarter, W.B., "An Overview of Artificial Intelligence and Robotics," NBSIR 82-2479, National Bureau of Standards, Washington, DC (1982).

Goto, 1974. Goto, N., *IEEE Trans. Electron Dev.* **ED-21** p. 662 (1974).

Harmon, 1980. Harmon, L.D., "Touch Sensing Technology: A Review," Society of Manufactoring Engineers (1980).

Kanade, 1983. Kanade, T., *Proceedings IEEE* **71** p. 88 (1983).

Lion, 1969. Lion, K.S., "Tansducers: Problems and Prospects," *IEEE Trans. Industr. Electron. Contr. Instrum.* **IECI-16** pp. 2-5 (1969).

Lozano-Perez, 1983. Lozano-Perez, T., *Proceedings IEEE* **71** p. 821 (1983).

Lubszynsky, 1978. Lubszynsky, A., *Adv. EEP* **64** p. 278 (1978).

Reddy, 1983. Reddy, R., *Proceedings IEEE* **71** p. 787 (1983).

Stanton, 1983. Stanton, M.D., Hill, H.S., and Crisp, N.D., "Grippers-Sensors and their Control Aspects," Proceedings of the 6th British Robot Association, Birmingham (May 1983).

Weimer, 1983. Weimer, P.K., *Adv. in Image Pickup and Display* **6** p. 177 (1983).

Wobschall, 1979. Wobschall, D., *Circuit Design for Electronic Instrumentation,* McGraw-Hill Book Company, New York (1979).

Woolvet, 1983. Woolvet, G.A., "Digital Transducers," in *Instruments Science and Technology*, ed. B.E. Jones, Adam Hilger Ltd, Bristol (1983).

Sanderson, 1983. Sanderson, A.C. and Perry, G., *Proceedings IEEE* **71** p. 856 (1983).

8. Computer Networks and Computer Integrated Manufacturing

8.1. Introduction

The rise of computer networking during the last years made it possible to connect computer systems in a highly reliable way. As a consequence of this, computer networks can now be used in Computer Integrated Manufacturing. Clearly, the concept of Computer Integrated Manufacturing would be inconceivable without computer networks.

The major task of the network is to take care of the integration of the various computer systems. By this, computer networks will contribute to high reliability, cost reduction and high flexibility.

In computer integrated manufacturing reliability is considered to be a most important design issue. The cost of a failure of the computer system can be prohibitively high, for example owing to production losses. The consequence of production losses should not be underestimated: their cost can be higher than the cost of a complete, integrated computer system. Reliability can be improved by using computer networks. To achieve this, the system should be built in such a way that when one computer breaks down its task will automatically be performed by another one. Clearly, an improvement of reliability implies cost reduction, even when extra investments are needed.

The use of computer networks contributes to cost reduction in more ways. Trends in hardware development lead to a better price performance ratio for small computers than for large ones. As the development time for small computers is short, later (and therefore cheaper) technology can be used. This results in networks of mini- and microcomputers more and more becoming economically justifiable.

The advantages of highly flexible manufacturing systems are well known. The most important one is that a system of high flexibility makes it easier to realize wishes of customers, wishes that are of a miscellaneous nature. As the production process can easily be influenced with the aid of computer networks, it will be easy to tune the production to the demand.

8.2. Communications strategy: an introduction.

In this section the relevant aspects of computer communications in CIM are defined, ultimately leading to the communications strategy. As the processing strategy is a method to use Information Technology to provide the best processing support for CIM, the communications strategy is a systematic method to provide the best possible computer communications support for CIM modules.

Important demands imposed on the communications strategy are:

- It should state how to build communication systems for CIM from a limited set of basic systems.
- It should state which communication protocols are applicable in CIM.

The first group of design rules follows from the first demand. To state these design rules one should carefully consider basic components of computer networks, like the transmission medium and the available hardware for low level protocols and judge how far these components, often not developed for use in CIM, are applicable in computer integrated manufacturing. It should be considered how requirements and constraints that are specific for CIM influence the choice of basic components.

This set of design rules will primarily deal with the communication requirements of "global" CIM modules as opposed to the second set, dealing with CIM submodules. The second set of design rules can be viewed as a set of "higher level" design rules. Starting from the fact that the basic communication requirements can be fulfilled by application of design rules from the first set, the second set states what high level protocols should be developed in order to make efficient use of the transmission facilities.

The use of standardized protocols will be emphasised; however, different applications will need different protocols of the top layers of the ISO-OSI model (ISO/TC97/SC16/N227, 1981). It will be investigated during the second part of the research project how specific requirements of communication inside CIM submodules will influence the high layer protocols.

In order to identify communication requirements in the various CIM modules, a list of characteristics of communication systems is given in figure 8.1.

The purpose of this list is to enable an easy classification of specific communication requirements for a particular CIM module. By this, it will be made easy to find adequate communication support for every CIM module involved.

8.3. Communication types associated with CIM-modules

For each CIM module the communication requirements need to be classified. The first question, a very important one, is what is the type of the required communication, for example does it need high data rates, is the communication bursty, what are the reliability constraints etc. The answers to these questions will sometimes make the use of certain types of networks impossible, for instance because they do not support sufficiently high effective data rates.

Next, environmental issues need to be considered and other issues related to the circumstances in which the communication takes place. To give some examples:

- The environment of the communication network can influence the choice of the transmission medium, especially when electrical noise is generated frequently
- The real time constraints of the communication will have influence on the choice of the hardware
- The size of the network will have immediate consequences on performance

These characteristics all have their own relevance. The implications for networking of the most relevant characteristics should be analyzed first. If these implications do not uniquely imply certain design decisions, less relevant characteristics will help in finding good and efficient network solutions. These solutions need not necessarily be unique; when there are no special requirements for a specific CIM module, many existing networks will satisfy the conditions, but, on the other hand, sometimes the requirements completely prescribe the network.

If the communication type and the "environmental circumstances" are known it will be easy to identify which combination of basic communication elements will suit the application best. This leads to a deliberate choice of these elements. Ultimately, the communication strategy will lead to an adequate support of communication in computer integrated manufacturing.

CHARACTERISTICS OF COMMUNICATIONS

Basic elements of communication.
- transmission medium
- topology
- bandwidth
- protocols
- reliability
- security
- expandability

Types of data transfer
- files
- pictures
- alarm messages
- short messages e.g. data from sensors
- voice
- video

Transmission media
- coaxial cable
- optical fibre
- satellite
- radio
- twisted pair cable

Local network designs
- Carrier Sense Networks
- Token ring
- Slotted ring
- Dual rooted tree

Protocols
- OSI architecture
- existing *de facto* standards
- inter network protocols
- high level protocols

Reliability
- requirements
- methods
- "upper limit" for message delivery time
- maintainability
- mean time between failure
- mean time to repair

Figure 8.1

It needs to be emphasised that the implications of the requirements of global CIM modules will be principally in the area of the hardware that can be used. When the sub-modules of CIM (e.g. a geometric modeller) are considered into more detail this will certainly influence the used communication protocols, especially those of the higher layers of the OSI model. During the second part of the research project it will be carefully investigated as to what demands are put on the protocols used and which standards will be applicable to computer integrated manufacturing.

8.4. Characteristics of communications in computer integrated manufacturing

In this section a description of communication in computer integrated manufacturing is given. The purpose of this classification is to simplify the recognition of important constraints that are imposed on computer networks used in this field.

A global classification of communication in computer integrated manufacturing consists of three parts: the first about communication during the design phase, the second about communication in manufacture and the third about communication occurring in both areas.

First we give a definition of **logical networks**.

A logical network is a distributed system, consisting of processors, software and physical network, designed to perform a specific task.
For each node of the network it is specified to which of the other nodes transmission can take place.

8.4.1. Design

Communication during the design phase, including computer aided design (CAD), computer aided engineering (CAE) and computer aided testing (CAT) is of a miscellaneous nature. Three important types of communication can be distinguished:

1 The transmission of data that enables an eminent interaction between designers, engineers and their work stations.

2 The transmission of data related to the use of distributed data bases.

3 The transmission of data related to analysis, the finite element method, the result of the design, etc. They are taken together because they can be characterized by the large amount of data involved.

The first type does not have real time constraints: it is not necessary to design the communication network in a way that precludes late arrival of messages. However, the speed with which results are obtained is important, because engineers and designers will be more inspired when the response time of the system is short, leading to a higher productivity. As a matter of course, this is not only related to the communication strategy, but also to the processing strategy, because when a good response time is required, it is necessary that information is produced fast, and, when this information becomes available, it is transported fast to its destination. As a consequence the network should have a high capacity.

The ever increasing integration in computer integrated manufacturing results in an increase of communication in the design phase, especially the transport of data out of huge data bases is needed. The use of distributed data bases has an enormous influence on the communication strategy. As long as these data bases do not change it can be advantageous to have several copies of them because this largely decreases the volume of communication required. The consequence of having several copies is, however, that updates of the data base require a lot of communication because all copies need to be updated as well. Apart from this, the problem of having multiple updates on multiple copies at the same time is not solved yet, and needs further investigated.

If every node of the network knows exactly where data can be found it can decide where a data base query has to be processed, based on this information as well as on information about the network load. If it not known in advance where data bases are maintained, broadcast messages to investigate where particular data is stored are needed. Problems arising from this design should be investigated further.

The communication between a CAD system and a CAE system consists of the transmission of large amounts of data, for instance solid models and sculptured surfaces. There are no real time constraints but it is necessary that the transmission delay is as small as possible. Given the amounts of data this requires a high performance network (one of which the effective data rate is high). The reliability of this network is not as important as of the shop floor network. Because of the gargantuan amounts of transactions the overall performance of the network should be good. If the network fails the failing part should be detected quickly and replaced without causing the network to fail completely.

8.4.2. Manufacture

In manufacturing three types of logical networks can be distinguished:

1 The control network

2 The monitoring network

3 The management network

The control network is responsible for the driving of machines, robots etc. In general, messages will be short but very important. As an example a description of the communication between robots and vision systems will be given. If a robot is used to acquire parts, it needs information about the scene that the vision system can see. More to the point the robot has to know the position of the part to be acquired exactly. Although this can be handled straightforward, currently no standard is available for communication protocols between robots and vision system. This situation is undesirable because it prevents integration of systems of different manufacturers. The introduction of a standardized communication protocol is made difficult by the existence of many already commercially established, incompatible robot languages and languages for vision systems. Often it is easiest to instruct a robot in its "own" language. If a standardized communication protocol is used, this would not necessarily contain the special instructions for a particular robot, implying that the messages should be translated or interpreted, which can slow down the effective communication rate largely. One might expect, nevertheless, that when a standard has been developed, manufacturers of robots as well as vision systems will adapt the languages they use, to make them better suited to the communication standard. It can be concluded that standardization is necessary, but made difficult by the number of languages already available.

It is evident that the arrival of wrong data in manufacturing can lead to disastrous consequences. For instance, when a moving crane is not slowed down in time, it can devastate a great part of the machinery on the work floor. Reliability thus will be a key issue in this type of communication. However, it is not possible to separate reliability and performance completely, because they are closely related. The reliability of a real time system is dependent on timely delivery of exact data. The timely delivery depends on the performance of the network under various loads as well as on its sensitivity to transmission errors, whereas the correctness of the data is related to the protocols used.

It is important to note that a gain in flexibility of the manufacturing increases the need for communication; sometimes this will influence the choice of the physical network. To conclude, this type of communication requires a high performance, reliable local network.

The monitoring network guards the correct functioning of every subsystem. To achieve this, it collects information from the control network. Several conditions should be satisfied.

1 Input material should always be available. Often this material will be produced by other subsystems, which implies that these subsystems should produce sufficient quantities to keep their 'successor' going. The monitoring network should detect when raw material is needed and act accordingly.

2 Machines and instruments should be in good repair. When it is detected that instruments do not function as required, the network should monitor this and act appropriately.

3 The monitoring network should establish that every subsystem meets its production demands. If production losses are inevitable this should be logged. In this way it will be easy to design a global system that takes care of the total production line.

In general, communication in a monitoring network will be local. It follows from the requirements that a real time network is needed.

The third type of network in manufacturing is the management network. Its main goal is to optimise the functioning of the production line as a whole. To achieve this, the output of the monitoring network of each subsystem should be inspected in detail, for instance to detect which subsystem is a bottleneck of the production line. Moreover, it is necessary to know what percentage of the production is lost in a particular subsystem. The management network collects this information from the monitoring network and acts appropriately when certain production units are performing badly. Production planning can be done easier when conscientious information concerning the current production is available. When the production is carefully analyzed it can be seen easily whether the requested production has been achieved or not and how the production planning has to be adapted to become reliable.

In contrast to the earlier logical mentioned networks, the management network will have to operate correctly during long periods of unattendance, namely as long as no information is needed. Of course it should always collect statistical and monitoring data and provide that this data is available immediately when needed. If the network is not working properly, this will, in general, not immediately be noticed. This implies that the network ought to be 'self-checking' that is it should have facilities to establish that failures of parts of the network can be detected and corrected. In this way, no human intervention will be needed to ensure that the network is functioning correctly. Moreover the network should have facilities for automatic rerouting etc.

8.4.3. General

A type of communication needed in both the design phase and the manufacturing phase is the communication related to the maintenance of the network. Even when networks are built in an extremely reliable way, it is unavoidable that errors in hard- and software occur.

These errors need to be detected quickly so that appropriate action can be taken. Often it will be easier to keep the network in good repair when it is possible to plug terminals into the network, for instance to detect which node is malfunctioning. These possibilities can be used best in networks without real time constraints.

8.4.4. Conclusions

It can be concluded that each type of logical network needed has its own requirements and responsibilities. There is also a clear need for communication between logical networks.

If the requirements of a logical network should not be violated against, it is necessary to minimize the mutual communication between logical networks.

Primarily in networks with real time constraints it will be easier to meet the obligations when the performance of the network is minimally affected by inter network communication. But speaking more generally and in terms of other networks it seems unnecessary and even unwise to let the inter network communication become large.

To give an example how the communication between networks can be minimized, consider the data flow between monitoring network and management network. Starting from the principle that normally the production system will perform well, the need for data transmission can be decreased largely when the monitoring network only reports production losses to the management network, so that when all is going well no communication is needed.

Furthermore, the requirements of each logical network are different. The differences are so large that it is unlikely that all requirements can be fulfilled in one existing (physical) network. At least nowadays it not possible to meet all requirements in one physical network and it is not likely that this situation will change in the near future.

Therefore communication in computer integrated manufacturing should make use of mixed technology (i.e. various physical networks).

Even for each individual logical network the technology is not necessarily prescribed. Sometimes more feasible solutions to the communications problem exist, and the choice will depend on other characteristics. When these characteristics are considered into detail one can choose the network that will suit the application best, depending on reliability constraints, performance constraints, etc.

References

ISO/TC97/SC16/N227, 1981. ISO/TC97/SC16/N227,, "Reference Model of Open Systems Interconnection," *Computer Communications Review* **11**(2) pp. 15-65 (April 1981).

9. Communication facilities

In the following sections an overview of the state of the art in computer networking is given, with the emphasis on the aspects that are of major importance to computer integrated manufacturing.

9.1. Network architecture

The architecture of modern computer networks is a seven-layer structure (see Figure 9.1).

Layer	Protocol	Layer
Application	.....	Application
Presentation	.....	Presentation
Session	.....	Session
Transport	.....	Transport
Network	.....	Network
Data Link	.....	Data Link
Physical	.....	Physical
Transmission medium		

Figure 9.1
Open Systems Interconnection

This structure has been agreed upon by the International Organization of Standardization (ISO/TC97/SC16/N227, 1981), and is commonly known as the ISO-OSI model (OSI = Open Systems Interconnection). Although many existing networks do not follow this standard completely, the concept is extremely attractive. The purpose of each layer in the OSI model is to offer a set of functions to its "upper neighbour" and to conceal the problem of implementing these primitives on top of lower layers. The higher layer has to use the functions of the next lower layer only. As a matter of course, the higher the layer, the more elaborate the functions it offers. The inter-layer interface has been designed carefully, so that changes that enable a better implementation of some layer (e.g. technological improvements) do not affect the implementation of higher layers.

A short description of each layer will be given now. More details can be found in Tanenbaum (Tanenbaum, 1981) and (Davies, 1979). Layer one, the physical layer, is concerned with the transmission of bits over a transmission channel. The meaning of the bits is of absolutely no concern to this layer. Layer two, the data link layer, adds the possibility of sending data frames over the network. Error checking is done, so that transmission errors cannot be seen by the next layer, called network layer. Its task is to divide data packets into (level two) data frames and to prevent too many data packets from using the same transmission line, causing congestion. The next layer is called transport layer. It accepts data from its upper neighbour layer, splits it into packets, and ensures that these packets arrive at their destination in the correct order. The transport layer is a real source-to-destination layer, as opposed to lower ones. In layer five, the session layer, connection establishment as well as connection management is performed. In this way the functions offered by the transport layer can be made more application oriented. The sixth layer, called presentation layer, has as main function the reduction of the amount of data sent over the net. Data compression usually will be done in this layer. Another function of the presentation layer is data encryption, a topic that is extremely important in long haul networks. The top layer is called application layer. It is up to the user to define protocols in this layer.

9.2. Transmission - analogue versus digital

The world of data transmission can be divided into two groups: analogue and digital.

Historically, analogue transmission, in which some physical quantity continuously varying in time is sent over a transmission line, has been most important. As communication between computers becomes more and more important, digital transmission is taking over this leading position.

The difference between analogue and digital transmission is that in analogue transmission the precise value of the signal is used, whereas in digital transmission the value of this signal will be viewed as being zero or one. Digital transmission has several advantages over analogue transmission (Tanenbaum, 1981). As the set of possible values to be send comprises only two elements the error rate is potentially low. A weakened signal can almost always be recognised, restored to its original value and repeated accordingly. Apart from this, several types of data can easily be multiplexed over one transmission line.

It should be weighed carefully in what form data will become available and in what form it should be transmitted. When digital data is transmitted in an analogue form, it has to be converted by a modem (modulator-demodulator). Also, when analogue signals are transmitted digitally, they should be converted by a codec (coder-decoder).

9.3. Transmission media and their characteristics

The purpose of computer networks is to transmit data over a communication channel. The hardware chosen largely influences important system parameters like performance, reliability and maintainability. Several transmission media are used; a characterization of each of them will be given in the following sections.

9.3.1. Coaxial cable

A very popular transmission medium is coaxial cable. Among its advantages are a low error rate, high data rates and low cost. The error rate varies from one bit in 10**7 bits to one bit in 10**11 bits. In extant local networks, the transmission medium is essentially error free.

As always in the real world, there is no advantage without an accessory disadvantage. Here it is the sensitiveness to electrical noise that takes care of a large increase of the error rate in environments in which such electrical noise is generated. It usually will be hard to isolate the cable sufficiently.

Coaxial cable can be used to support two sorts of transmission channels: baseband and broadband.

9.3.1.1. Baseband technology

When a coaxial cable provides a baseband channel, digital data is directly put onto the cable; the cable is used as a single channel. Digital information departs from the transmitting node in both directions. The transmission rate can be up to 50 Mbps, depending on the distance that has to be covered. The great advantage of the coaxial cable is the ease of installing or removing nodes from the network, often without disturbing current network traffic. As distinct from optical fibres, coaxial cable can be used in many network configurations, not only in point-to-point networks.

9.3.1.2. Broadband technology

Broadband-based networks use a technology that is widely used for cable television (CATV). The physical channel is divided into several separate logical channels by use of Frequency Division Multiplexing (FDM). Broadband cable television uses directional broadcasting, putting some constraints on networks using this technology. A solution to the problem of having directional transmission only, whereas bidirectional is needed is to let one node act as a repeater: it has to repeat the incoming data stream in a different frequency range (to preclude interference). The consequence of this policy is that a message generally has to go farther than with bidirectional transmission: first to the repeater node, then back to the destination which means that transmission delays are increased. The advantage of broadband networks is that multiple logical channels can be supported, each with a different bandwidth. This means that all kinds of data (voice, video, digital data) can easily be multiplexed on one physical channel.

9.3.2. Twisted pair cable

Twisted pair cable technology is frequently used by telephone companies. Depending on wire length it can handle frequencies up to 10 KHz. Its main advantage is its low cost, its most important disadvantage is its susceptibility to external interference. This means that twisted pair cables work well for the transmission purposes they historically were intended for: telephone calls and other low frequency applications. However, they are totally inadequate as transmission medium in a high performance, high reliability network.

9.3.3. Optical fibres

Optical fibres have several desirable properties. First, the transmission rate can be very high (up to 10**9 bits per second). Secondly, the error rate is low: about one error in 10**9 bits. Thirdly, this error rate does not increase in hostile environments: it is not influenced by radio frequency interference, electromagnetic interference or electromagnetic pulses. In contrast to conventional communication media crosstalk is negligible, even when many fibres are cabled together. Another characteristic of optical fibres is that they are best suited to digital transmission. Additionally, fibre optic cables are thin and have a low weight, which makes them easy to install.

But there are also some disadvantages. The most important one is the difficulty to retransmit an incoming signal on multiple output lines. This implies that optical fibres are best suited to point-to-point networks, that is networks in which each node has exactly two neighbours. Furthermore, it is difficult (or almost impossible) to add additional nodes to the network, implying that the topology of the network should remain constant during its operation, providing for less flexibility. Other disadvantages are the lack of standardization of components and, given the current state of the art, the fact that fibre optic technology is expensive. However, as soon as fibre optic technology will be used more, prices will rapidly fall and this will make the new technology cost effective.

Now that the specific advantages and disadvantages of fibre optic technology are collected, it is time to become more specific and make conclusions from this information. Fibre optic technology is superior to other transmission media in several important ways: principally in networks where noise will be generated the reliability of the network can be increased when fibres are used. Since the cost are still high, it should be weighed up carefully whether the gain in reliability is worth the extra cost. As long as it is impossible to install or remove nodes from the network without disturbing network traffic, fibre optic technology can be used best in networks where the topology is stable for relatively long periods.

For these kind of networks, optical fibres will soon be the best available transmission medium.

9.3.4. Satellites

Communication satellites can be seen as repeaters. A signal that is sent to it will be broadcast in another frequency range, so that other earth stations can receive it. Satellites that are used for transmission purposes are usually at an altitude of approximately 36,000 kilometers, so the time that elapses before the transmitted data can be received will be substantial: 270 msec. Apart from this, the error rate of a satellite channel is dependent on meteorological conditions. Protocol designers should take error rates of one in 10**4 bits into account.

Satellites can be of great importance in long haul networks, especially when much of the data should be broadcast. Often the only alternative way of communication is to use the existing (low-bandwidth) telephone system. When large amounts of data have to be sent satellites can perform better. However, they are not suited well to local networks mainly because of the large propagation delay.

9.3.5. Radio

Radio channels can be used in two different ways, one being broadcast, the other point-to-point. The data rate that can be achieved depends on the frequency of the radio signal. For reasonable data rates this implies frequencies in the MHz or GHz range. The error rate of radio channels is higher than with other transmission media. This means that radio channels are not particularly suited to communications in computer integrated manufacturing. A possible use of radio channels, however, can be the connection of two cable based networks, for instance because the distance between them is large and it presents difficulties to use another transmission medium.

9.4. Local Area Networks

Some characteristics of local area networks are (Franta, 1981, Tanenbaum, 1981):

1 They are small, the size of the of the network does not exceed more than a few tens of kilometers; usually a local area network is contained in a complex of buildings.

2 They are usually owned by a single organization.

3 They have a data rate of more than 1Mbps (Million bits per second).

As these three conditions will generally be satisfied in computer integrated manufacturing, the importance of local area networks is elucidated.

9.4.1. Ring Networks

Ring networks are called this, because their topology is shaped like a ring. Every node of the ring has a point-to-point connection to two neighbours, except for the trivial case of a network of two nodes. The connections can be made by using coaxial cable, twisted pair cable or optical fibres. Since a ring consists of point-to-point connections, one can choose the most appropriate transmission medium for each connection separately.

9.4.1.1. Token rings

The most important type of ring is the token ring. When the ring is idle a bit pattern, called the token, circulates around the ring, from node to node. The token performs the duties of a permission to send, which means that a node can send only when the token has just passed by. To prevent other nodes from transmitting, before sending is actually started the last bit of the token is inverted. The hereby generated bit pattern is called a connector. To prevent token and connector to appear in the data, bit stuffing will be used. The concept of bit stuffing will be explained by an example. If the token consists of eight 1 bits, the sending node will generate a 0 bit after seven succeeding 1 bits in the data, whereas the receiving node will remove this 0 bit. By this, it is precluded that tokens appear in the data on the ring even when they appear in the original user data. The data transmission will begin with the transmission of a connector, after which the actual data is sent, followed by a new token.

9.4.1.2. Contention rings

A disadvantage of a token ring is the delay introduced by waiting for a token under conditions of low load. Because of this a new ring design was suggested, the contention ring. On a token ring a token circulates even when there is no traffic, but a contention ring is quiet in these circumstances. Eventually, a node decides to send, it puts its data on the ring, followed by a token. Other nodes can monitor the traffic on the ring and wait for the token before they transmit. This means that when the contention ring is full of packets it acts like a token ring. The only problem is that two nodes might both decide to start data transmission because they both think that the channel is idle. In that case a collision occurs and both have to wait a random time before their new attempt.

9.4.1.3. Slotted ring

Another kind of ring is the slotted ring or empty packet ring. In this ring a number of fixed size packet slots circulates around the ring, each with a leading bit to indicate whether the slot is empty or not. A node that want to transmit waits until an empty packet slot arrives, marks it as full and puts its data in it.

A disadvantage of this system can be that each packet has the same size. When this size is small, messages have to be divided into several packets, if it is large, bandwidth is waisted because packets will not be filled completely. Furthermore, sometimes artificial delays are needed in the repeaters, because otherwise the ring cannot contain all packets simultaneously. This can be caused by a large number of packet slots, a large packet size or a small number of nodes.

A well known slotted ring is the Cambridge ring (Wilkes, 1979). Packets consist of two bytes of data. The first implementation of the ring had several twisted pair connections as well as one fibre optic link. Nowadays they are often used in many network areas.

9.4.1.4. Problems in the design of a ring

This section is based on an article by Saltzer et al (Saltzer, 1983). The main problems in ring design are the reliability of the repeater string, distributed initialization and closed loop clock coordination.

In the basic token ring a failure of one node can cause the failure of the entire network, since all nodes are active repeaters. A possible solution is to let every inter node transmission line go through a central point, the wire center. At this center, bypassing schemes can be used to disconnect failing nodes or lines. The resulting configuration, a star shaped ring, is commercially available.

The problem of distributed initialization is also solved (Saltzer, 1981). As we have seen in the previous section the ring is in trouble when the token is lost. Although the probability of transmission errors in Local Area Networks is usually low, it is not zero and therefore the aforementioned problem can occur.

To handle this problem a strategy has been developed. It consists of two parts, one that detects the trouble, the other solves it. Discovery of token loss can be done by a simple timer. When token loss is detected, a special, so-called jamming signal is put on the ring to ensure that every node knows about the current situation. Then, every node tries to reinitialize the ring, one after the other, based on each node's station number. If an attempt fails, for example owing to collisions or transmission errors, the same node is prohibited to try again. This simple scheme guarantees that, with a high probability the ring will be functioning soon after token loss.

9.4.2. Carrier sense networks

The transmission media most suited to carrier sense networks are coaxial cable and twisted pair cables. In contrast to the token ring, the topology of the net is not prescribed in a carrier sense network. It can be linear, tree, segmented etc. The basic point of carrier sense networks is their access protocol. It is based on the fact that the propagation delay (i.e. the time necessary to reach the farthest node of the network and return) is small compared to the packet transmission time. This means that a node can listen whether other nodes are transmitting, and will try to make access with the channel only, when it does not hear any transmissions. Although, because of the propagation delay collisions cannot be prevented completely, this protocol performs better than a protocol without the listen-before-transmit-feature.

The most widely known carrier sense network is Ethernet (Metcalfe, 1976). [Ethernet is a trademark of Xerox Corporation] Its transmission medium is coaxial cable. The access is based on the above-mentioned listen-before-transmit-feature. Whenever a collision is detected, the stations involved stop their transmission, wait some time, different for each station and depending on the number of failed transmissions, and later try to access the net as usual. This protocol is called CSMA/CD for Carrier Sense Multiple Access with Collision Detection. Ethernet is a most commonly used local area networks.

9.4.3. Dual rooted tree

A local network design that may become important in the future is the dual rooted tree, developed at the University of Toronto (Boulton, 1983). It uses glass fibres as its transmission medium and, as its name suggests, its topology is a dual rooted tree. The access protocol is relatively simple but requires that a node can listen to all its incoming transmission lines simultaneously and handle accordingly. Under these conditions the performance can be very high. However, the dual rooted tree is merely a research project, not involved in standardization efforts of the IEEE 802 committee (Graube, 1982, Clancy, 1982).

9.4.4. A comparison of Ethernet and token ring

The most commonly used local area networks nowadays are carrier sense networks and token rings. Because of this, many comparisons of the two have been made (Chanson, 1982, Pogran, 1983). We will try to summarise the results, emphasising arguments that can be important in Computer Integrated Manufacturing. Several characteristics of local area networks are of interest: performance, reliability, maintainability and fairness of the access protocol.

9.4.4.1. Performance

Intuitively, a carrier sense network performs well, when it is lightly loaded because a node can send immediately when it wants to, and the probability of collisions is low. On the other hand, in a highly loaded carrier sense network the likelihood of collisions is high, which means that the packet delay can be high. In comparison to a carrier sense network with CSMA/CD a token ring will perform worse under conditions of low load, because a node has to wait for the arrival of the token, but performs better under conditions of high load, because it is essentially collision free.

In a carrier sense network the performance is highly dependent on the ratio propagation delay versus packet transmission time. When this ratio is high, for example because of the large size of the network, or because of the high data rate of the transmission medium, the advantage of the collision detect mechanism disappears almost completely. This implies that when better transmission media become available, the performance of a token ring will improve more than the performance of a carrier sense network.

A more formal approach to performance can be found in (Bux, 1981), and (Stuck, 1983). The performance of an Ethernet is described in (Shoch, 1980). Tobagi and Hunt analyse the performance of an CSMA/CD network (Tobagi, 1980).

We need to emphasise that the policy of using highly loaded networks is not considered to be a good one, especially not in networks with real time constraints.

9.4.4.2. Reliability

Reliability is considered to be one of the most important issues for comparison. Essential is that a failure of one node of the network will not cause damage to the rest of the network. In the earliest token ring designs, such a failure would disconnect the network, a highly undesirable situation. Star-shaped rings however (Salwen, 1983), do overcome this problem, carrier sense networks do not have it.

Another point of interest is the failure rate of hardware and software. As Graube puts it (Graube, 1982) there should be only one undetected transmission error per year. In Computer Integrated Manufacturing there can be situations in which even this error rate can be too high. But, in a hostile environment like in some factories, the probability of transmission errors can be high due to electromagnetic induction, unless special precautions have been taken. The most elegant solution to the problem is to make use of optical fibre technology. An adventitious advantage is the possibility of high transmission rates. At this moment the problem of turning a broadcast network into a fibre optics network is not satisfactorily solved (Saltzer, 1983). Token rings can easily be adapted to optical fibres.

A point worth noting is the desired upper bound on the packet transmission delay. Unfortunately, given the present state of the art, there is no guaranteed upper bound. In both networks designs transmission errors can ruin the data, leading to retransmissions. Moreover in an Ethernet there is a (low) probability of repeated collisions, whereas in a token ring token loss, owing to transmission errors, and resulting reinitialization procedures can theoretically increase the transmission delay considerably.

9.4.4.3. Maintainability

It should be easy to repair failing connections and nodes. It should be easy to add new nodes to the network and to remove unused nodes. When the topology of the network is changed this should not influence the normal operation of the network. In both the star shaped ring and an Ethernet it is easy to remove a failing node from the network. Detection of the failing node is easier in the ring (Saltzer, 1983).

9.4.4.4. Fairness of the access protocol

In the token ring a fair access protocol is provided, that is a protocol in which no particular node can sabotage the entire network by sending packets continuously. In a broadcast network with CSMA/CD the situation is different. Because in most systems the time a node waits before retrying after a collision is positively correlated to the number of collisions a packet has suffered from, a packet that has been involved in several collisions is discriminated against.

In both network designs, it is not easy to have priorized packets. In Computer Integrated Manufacturing this is an undesirable situation, because there are excellent applications of these. One example is the use of alarm messages in a factory. However, as transmission media become faster, the need of high priority messages will probably become less important.

9.4.5. Local networks and standardization

When the International Organization of Standardization started its network standardization efforts, leading to the model of Open Systems Interconnection, this model was designed with a view to long haul networks. Local networks differ from long haul networks in many ways and this finds expression in the way the physical and data link layer of local networks are designed. Although local networks do not fit in the OSI model completely, some standardization was clearly needed. The IEEE 802 committee has the intention to provide standards for local networks. Layers above the data link layer of OSI are beyond the scope of these efforts, which means that local networks and long haul networks adhere to the same standards from the network layer.

As indicated already, several incompatible schemes for local networks exist. Therefore the draft of the IEEE 802 standard distinguishes three different types: contention networks with CSMA/CD, token rings and token busses. For each of these types a standard will be created.

9.5. Wide Area Networks

In computer integrated manufacturing wide area networks are not as important as local area networks. However, for some applications they play an important role, for instance the research and development department of an industrial organization will benefit largely from the use of wide area networks, because their use gives the opportunity to react adequately to developments in the "outside world". It is evident that the use of wide area networks in this way gives enormous advantages, but, on the other hand, this implies serious threats to the security of the integrated computer system. The most secure way to ensure that researchers of other organizations will not copy (or modify) private information is to let a special computer serve as an "information-conveyor". Information can be brought to this special purpose computer from other in house computers (for instance via a local network) and from the outside world, but it should not be possible to reach the in house computers via this computer. Still, special precautions have to be taken to ensure that information can be used only by the people it is intended for.

Another field where wide area networks are used is as connection between several geographically dispersed local area networks. By this, integration of the complete computer system can be accomplished. Since often the telephone system is used to achieve interconnection, protection and security problem need to be considered carefully.

9.6. Reliability of networks in Computer Integrated Manufacturing

Reliability of the networks is, as stated before, one of the key issues in computer integrated manufacturing. On the lowest layers of the OSI-model almost all transmission errors are corrected by the use of cyclic redundancy code, for which hardware is available. To satisfy the extremely high demands on reliability in computer integrated manufacturing more warrants are needed. These demands can be met by software in the high level protocols.

It should be noted that, especially when the number of different orders to a machine is small, transmission errors that are not corrected by the low layer hard- or software can easily be detected, because an arbitrary error will generally not produce a valid order to the machine. Therefore we can conclude that:

The interface to machines ought to be kept as small as possible.

This is an easy way to increase reliability without affecting the performance too much. When more guarantees are built in in higher layer protocols this will undoubtedly affect the performance of these protocols. It is clear that this performance must not decrease below a well defined level. If reliability is to be built in in high layer protocols, performance certainly is a constraint.

9.7. Protection and Security

In a fully integrated computer system, including a computer network the timely arrival of exact data is very important. To achieve this goal the network should not only be reliable but it is also necessary that a mechanism exists that takes care of protection against unauthorized access.

Networks used in computer integrated manufacturing should have their protocols built so that messages containing clearly wrong instructions should be recognised, discarded and reported to a central system manager. These protocols provide more security and are built on top of the protocols that take care of messages that are garbled during transmission.

Especially in networks that use public transmission media problems concerning protection and security can occur. According to Voydock and Kent (Voydock, 1983) threats to communications security can be divided in two categories, the passive (unauthorized release of information) and the active attacks (e.g. unauthorized modification of information). They conclude that passive attacks can be prevented but not detected, whereas active attacks can be detected but not prevented.

The protocols of the presentation layer should be designed carefully to withstand all security threats.

References

Boulton, 1983. Boulton, P.I.P. and Lee, E.S., "Bus, Ring, Star and Tree Local Area Networks," *Computer Communications Review* **13**(3) pp. 19-24 (July 1983).

Bux, 1981. Bux, W., "Local Area Subnetworks: A performance comparison," *IEEE transactions on communications* **COM 29**(10) pp. 1465-1473 (October 1981).

Chanson, 1982. Chanson, S.T., Kamur, A., and Nadkarni, A.V., "Performance of some local area network technologies," TR 82-8, University of British Columbia, Department of Computer Science (1982).

Clancy, 1982. Clancy, G.J. and Harrison, T.J., "Local Area Network Standards: A status report," pp. 149-161 in *Distributed Computer Control Systems 1981*, ed. W.E. Miller, (1982).

Davies, 1979. Davies, D.W., Barber, D.L.A., Price, W.L., and Solomonides, C.M., *Computer Networks and their protocols,* John Wiley and Sons, Chicester (Great Britain) (1979).

Graube, 1982. Graube, M., "Local Area Nets: a pair of standards," *IEEE Spectrum* **19**(6) pp. 60-64 (June 1982).

ISO/TC97/SC16/N227, 1981. ISO/TC97/SC16/N227,, "Reference Model of Open Systems Interconnection," *Computer Communications Review* **11**(2) pp. 15-65 (April 1981).

Metcalfe, 1976. Metcalfe, R.M. and Boggs, D.R., "Ethernet: Distributed Packet Switching for Local Computer Networks," *Comm. ACM* **19** pp. 395-404 (1976).

Pogran, 1983. Pogran, K.T., "Selection of a local area network for the Cronus distributed operating system," *Computer Communications Review* **13**(3) pp. 25-35 (July 1983 distributed operating system).

Saltzer, 1981. Saltzer, J.H., "Communication ring initialization without central control," MIT/LCS/TM202, Massachusetts Institute of Technology (December 1981).

Saltzer, 1983. Saltzer, J.H., Pogran, K.T., and Clark, D.D., "Why a ring?," *Computer Networks* **7**(4) pp. 223-231 (1983).

Salwen, 1983. Salwen, H.C., "In praise of ring architecture for local area networks," *Computer Design*, pp. 183-192 (March 1983).

Shoch, 1980. Shoch, J.F. and Hupp, J.A., "Measured Performance of an Ethernet Local Network," *Communications of the ACM* **23**(12) pp. 711-721 (December 1980).

Stuck, 1983. Stuck, B.W., "Calculating the maximum mean data rate in local area networks," *Computer* **16**(5) pp. 72-76 (May 1983).

Tanenbaum, 1981. Tanenbaum, A.S., *Computer Networks,* Prentice Hall, Englewood Cliffs (1981).

Tobagi, 1980. Tobagi, F.A. and Hunt, V.B., "Performance analysis of Carrier Sense Multiple Access with Collision Detect," *Computer Networks* **4** pp. 245-259 (1980).

Voydock, 1983. Voydock, V.L. and Kent, S.T., "Security Mechanisms in High Level Protocols," *Computing Surveys* **15**(2) pp. 135-171 (June 1983).

Franta, 1981. Franta, W.R. and Chlamtac, I., *Local Networks,* D.C. Heath, Lexington (Mass.) (1981).

Wilkes, 1979. Wilkes, M.V. and Wheeler, D.J., "The Cambridge Digital Communication Ring," pp. 47-61 in *Proceedings of the Mitre NBS Local Area Communications Network Symposium held in Boston*, (1979).

10. Future Trends

In chapter one already many possibilities were mentioned for increasing integration and flexibility of CIM systems. It is interesting to see that currently not all options for improvements are chosen. There seems to be a fairly consistent strategy which is being followed independently by CIM systems developers. The consistency mainly is about the silent agreement which options for further integration to choose.

In Appendix A an overview is given among other things about current and future trends for CIM in Japan. In a later stage a similar appendix will be added for trends in the USA.

In the following an outline is given about the influence or consequences these developments have with respect to processing strategy. It is not the intention to evaluate at this moment whether the options taken are to be considered the right choice. Probably this has to be done in a wider context, i.e., taking into account also data strategy and overall CIM architecture.

CAD/CAE: The most important trend in CAD is to take more design parameters into consideration. This results in a higher degree of integration, in particular, because many of the parameters now originate with the production process. The support for such a flexible design method is mainly provided by enhanced database functions and sophisticated simulation facilities.

Simulation is an important design aid. It requires powerful processors (number crunchers). To provide for this, interactive CAD/CAE workstations are connected via high speed links with mainframe type computers which execute the simulation tasks. The result can be shown in real time on the graphics workstation.

Following this trend some of the more frequently used simulations such as those providing realistic 3D pictures, are being built in the interactive workstation, with the help of custom designed VLSI implementations of the time consuming algorithmic kernel of such a simulation support package.

The high speed links between workstation and (remote) mainframe ask for powerful communication hardware as well. In addition, the simulators will use large, very likely distributed databases. These put an even more heavy demand on communication facilities.

FMS: The main emphasis in improving Flexible Manufacturing is currently on flexible transportation of parts and tools. On the basis of a thorough, company based standardization for pallets, working levels, etc., a large variety of parts can be handled in a flexible manner by part handling robots. These are either on board of an FMS cell, or they are part of the conveyor system.

Serious attempts are made to introduce on line programming of these robots, because of the considerable improvement in lead time which is to be expected from this. However, sufficiently sophisticated robot programming languages required for this are still under development.

Following the successful introduction of programming for part handling robots, whenever that may be, it is to be expected that programming for assembly robots will then be within reach too.

Much development will take place in sensor systems to be built in FMS cells. They will take care of monitoring tasks and the checking of machining and assembly tasks. Adaptive control mechanisms are being built in sensor systems for a variety of applications, such as:

- Feedback for multipass machining, allowing for very high accuracy.
- Compliance operations for assembly.
- Quality control of subassemblies.

All FMS cell functions are controlled by one processor system which provides all local processing support, e.g., for cutting, sensing and signalling to the outside world.

The interface to the outside world, i.e. the central computer of the shop floor is being standardized. It is envisaged that this architecture will also be the next generations architecture. It has sufficient resources and possibilities for adding more functionality.

This is a consequence of the philosophy to organize flexible manufacturing around powerful FMS cells, together with a very flexible transportation system.

CAD/CAE for FMS: The integration of CAD/CAE and FMS is considered to be the most difficult part, both for the designers, with or without computer support, and for the CIM system builders.

In all cases integration is achieved on the basis of a very powerful and sophisticated database system. Knowledge basis are considered necessary and effective to fully support this integration. Most likely artificial intelligence will be introduced in this area. It is currently difficult to predict how soon these systems can be used in practice. This will depend very much on the success of the fifth generation architectures.

APPENDIX A

The important characteristics of C I M developments in Japan.

By: P.J.W. ten Hagen &
L.O. Hertzberger.

B01/84

The important characteristics of CIM developments in Japan **

by

P.J.W. ten Hagen & L.O. Hertzberger *

A B S T R A C T

This paper describes the state of the art of Information Technology as applied to CAD and CAM in Japan. An overview is given of CAD systems, as well as their basic components, Graphics, Geometric modelling and Simulation. In the discussion of CAM emphasis is given to Flexible Manufacturing Systems, in particular to the design strategy for these systems. It can be expected that future CAD/CAM systems will make extensive use of knowledge based systems, whereas FMS will apply advanced robots and robot sensors, in particular robot vision.

Keywords and phrases: Computer Aided Design, Computer Aided Manufacturing, Flexible Manufacturing, Geometric modelling, Simulation, Robots, Sensors, Vision systems.

---

* Department of Computer Science, University of Amsterdam.

** This report will be submitted for publication elsewhere and will also appear as a report of the Mathematical Centre.

1.0 Introduction

This is a report of a visit to Japan to survey the current situation and trends in Computer Integrated Manufacturing (CIM) and related topics.
The visit was made as part of a research project in the EEG ESPRIT CIM program.
Special attention was paid to the reasons for the advanced state of the CIM situation in Japan as compared to Europe. Being active in the area of Information Technology (IT) only IT aspects of CIM are considered here.

In the industries visited the overall approach to IT strategies showed strong similarities. All industries attempted to have all the Information Technologies necessary for their design and production in house. They are prepared to undertake all R&D activities necessary to accomplish this goal. Alternatively they are willing to invest in buying the basics for such technology. They are even prepared to build their own special purpose hardware, including VLSI.
The industries have advanced CIM systems, which are based on current IT also present in Europe but maybe not that accessible. Our impression is that they use current technology better by simplifying it, which makes it cheaper and easier to integrate. Every company puts strong emphasis on standardization.
They consider integration as a step-wise process. In each process a number of strategic modules of the overall process are analysed and improved as well as integrated. For instance, they have concentrated on the automation of machining but not so much on assembly automation.

Besides these general impressions, important aspects were found, concerning specific CIM modules. These will be discussed in the next sections.

2.0 CAD for Mechanics components

Many of the CAD systems currently used in Japan are being imported primarily from the USA, such as the IBM CADAM system. Almost everywhere developments take place to design a next generation of CAD systems. An extra stimulus is the language problem associated with imported systems.
CAD systems are predominantly used in drafting systems, however, work is in progress to extend them to full 3D systems.
For a number of applications advanced CAD systems have already been produced. These systems are not marketed because they form part of the respective companies in house technology.
One example is the five year project which MITI has started and deals with the integration of CAD and CAM. They have designed a system that generates all cutting processes for NC machining from 3 sets of CAD data: the initial form of the rough material, the final product shape and a set of tolerance data. Another achievement was a new geometric modeler containing a 2-level hierarchy with a winged edge description at the lowest, and a volume gluing type of modeller on top. The integration between CAD and CAM is achieved via the database.

In industry demonstrations were given of a sheet metal handling system. The design drawings and the production flow through the factory were used as an input to the database. The sheet metal handling system was able to generate the commands for the metal press and the punching line. Most of the systems were using an integration method as illustrated in figure 1. From these examples it is clear that more and more production flow data is entering the CAD systems.

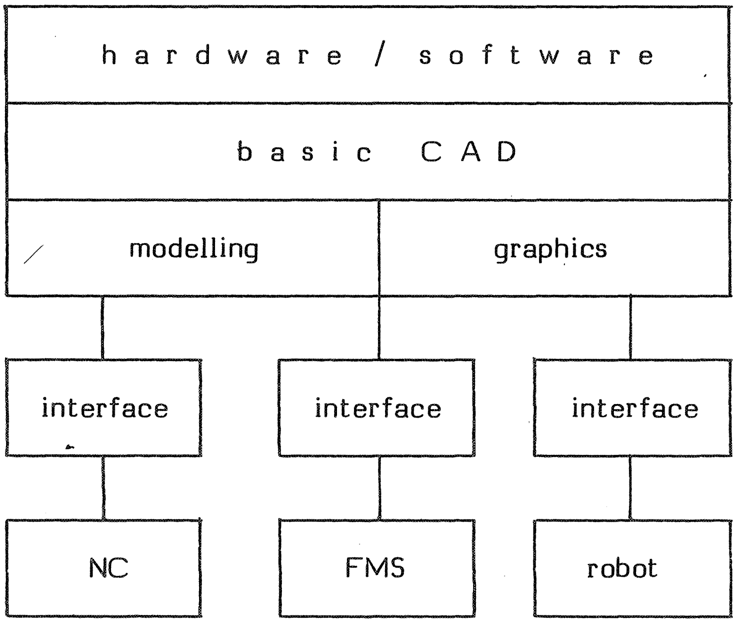

figure 1: CAD/CAM integration
(courtesy to FACOM)

This is even more true in the cases where flexible manufacturing is used.
At the University of Tokyo a CAD/CAM system was illustrated, the objectives of which were to design integrated flexible automation systems based on CAD/CAM systems using additional assembly data.
For that purpose the following equipment has been set up:

- graphics terminal for modelling and simulation
- automatic on-line wood cutter to cut models
- an automatic measuring table to measure models
- an NC control with modelling possibility
- a robot aimed to be used for measuring and production purposes

The CAD system is based on the modeller Geomap III. The purpose is to design an integrated system for assembly, machining, inspection and measurement as well as robot handling. Emphasis is given to basic programming of such a system but all developments take place by doing experiments with the on-line equipment. They consider it essential for the integration of all system functions that semantic data can be added to the database of the geometric modeller. Semantic data being information from all phases in the production process, and not only geometric but also physical, technological, and management data.

Because the process of engineering is well defined, it is foreseen to store engineering data in a knowledge base.

Other examples of the integration of other information, such as production planning, with the CAD/CAM systems are illustrated by an operational CAD system for car manufacturing.

The car manufacturing consists of the following processes
1. planning
2. styling design
3. body engineering
4. stamping die design
5. stamping die manufacturing
6. production engineering

1. In the planning phase a parts database is set up containing information about own and competitor parts. In this phase decisions concerning image, dimensions and costs are taken.

2. This phase starts off with a rough artist sketch that is digitized by tablet. Thereafter more detailed design data is brought in interactively. Then iteration loops involving smoothing, modifying and redesign take place. Also stiffness tests are applied. Finally when the shape is defined an on-line cutting machine produces a clay model.

3. First a 3D finite element mesh model is made of the carbody. This is used for structured force analysis. Stress control maps are produced for visual check of stress. In addition product drawings are produced containing dimension lines and numbers.

4. First stamping direction is determined and checked for correctness. Thereafter die and product surfaces are calculated. For stamping, stress analysis is performed.

5. NC control tapes for numerical cutting of moles are generated.

6. Robot welding and NC control specifications are produced.

The most advanced system studied was the so-called Total Design CAD system. In this system intensive simulation facilities were included in order to have realistic presentation of 3D design images.
The CAD system structure contained three integrated subsystems. A 3D modeller describing the designer's concept. Next the engineering stage where a large variety of kinematic simulation was performed. In this stage part geometry and the description for realistic presentation was used. This was all based on the 3D interactive modeller. In the third stage, the manufacturing stage, all data necessary for manufacturing was produced, such as cutting programs necessary to produce 3D models.
The man-machine interface of the CAD system was based on the new interactive technique being developed. For instance, 3D position input was supported by feed-back in multiple views (4). Also scanner input could be used of natural patterns that subsequently could be manipulated by the 3D modeller. The method for realistic presentation was developed in close collaboration with industrial design artists. It was able to do surface modelling with color reflection etc. In conclusion the 3D CAD system was aiming at an utmost realistic design environment capable of assisting the designer by visualising the interior of complex mechanisms, for instance, the animated simulation sequences with multiple viewing.

2.1 CAD for FMS

The situations where CAD is used to design products to be produced on an flexible manufacturing line, special emphasis is payed to the rationalization of the engineering process. This is very important as the number of tools that are used in flexible manufacturing lines have to be limited. Also the number of actions that can be performed by the robot based machines of the flexible manufacturing line are limited. One of the challenging problems for the future is the connection of CAD systems towards flexible manufacturing lines.
A CAD system was studied that was able to produce flexible manufacturing line planning based on custom specification of the products. Now the system produces planning charts. In a later stage the CAD system will be connected to the FMS cells and control them directly.
The CAD/CAM, in particular FMS, is necessary but by itself does not reduce lead time, this is because special tools and fixtures still have to be prepared.
Automation of NC-programming does not change this situation. The only solution seen is total rationalisation of CAM processes, which results in standardisation of tools and fixtures. Next these standards have to be input to the CAD system to obtain integration. An essential condition for using standards in CAM is that they have a user friendly interface especially for the designer. For instance, there is a need for an enhanced man-machine interface to interactive NC-programming, something like APT is considered useless in such an environment.
In conclusion the designer's freedom is reduced by forcing standards on him. There are still problems in this area. A solution might be found in building a CAD system with a man-machine interface allowing optimal use of the standards, i.e. a CAD tool which enables designing with standards, without hindrance to creative design.
Because of the standardization the number of machining tools went down from 300 to 72. The absolute maximum number of tools a designer is allowed to use is 80, equivalent to the maximum number a machine drum can hold.
After the installation of FMS all engineers had to be retrained and many of them had to be replaced.

The flexible manufacturing lines themselves can also be designed using a CAD system. This is possible because in designing products using flexible manufacturing, it becomes essential to understand the production lines themselves.
The robots along the FMS production lines can also be standardized with, as a consequence, the further rationalization of an FMS line. The further this line is rationalized the easier it becomes to design new FMS cells, based on parts of already designed cells. The uniformity also makes maintenance of the system easier. When a cell goes down, only a limited amount of spare components is necessary to replace the failing parts.

2.2 Graphics

Many 3D basic modelling systems are being developed. The basic graphics package used is the GSPC79 CORE. There is a great interest in the 3D extension of GKS.

The new 3D modellors are interactively used. This requires dynamic 3D movement for feed back, and real-time generation of shaded images with hidden surfaces removed. It is envisaged that highly powered graphics workstations and the computational power of on-line mainframes will be necessary to support this. At several places a mainframe dedicated to CAD work was available.
Some examples will be discussed to illustrate the power of various 3D graphics stations and their associated software.

A graphics device was studied, that was used for various CAD applications. The station was coupled to a supercomputer via 1.6 Mbyte/s DMA channel. The target applications were structured surface design, molecular simulation and numerical analysis. The supercomputer can produce a three dimensional picture with hidden line removal containing 1600 polygons in 10 seconds. The resulting image contains about 500k pixels. The 3D software uses a method of surface description interpolating over a large area with special coons patches. Another interesting feature was the so called median cut method to find the best color approximating out of a limited set.

A CAD system was studied, with special 3D facilities to be used for mechanical engineering. The man-machine interface, especially the command language and input techniques have been specified by the designers. The current system is the third generation CAD system in use. For each generation the design staff produced part of the specifications.
The system consists of about 20 high quality vector graphics workstations (AGS) derectly coupled to a large mainframe computer, completely dedicated to the CAD work. The application runs completely in the host. The workstation intelligence provides high level feed-back from input. For real-time picture change a structured data file is maintained. Feed-back functions include 3D real-time transformations, real-time curvature change under operator control and selection and definition of curved surfaces.
The system is one of the most advaced existing systems.
It allows for integration of CAD and production engineering. The CAD system developers are planning to base CAD development on the 3D extension of GKS.

2.3 Simulation

Almost every CAD system under development provided application oriented simulation techniques. The simulation calculations were done in the mainframe, the visualisations, sometimes in real-time, were supported at the workstation level.
The integration of simulation models in CAD is a typical example of stepwise realisation of fully integrated CIM systems.
Much attention was paid to the careful addition of simulation facilities towards the man machine interface.
An example of how simulation is used in complex 3D CAD systems was illustrated in the last example of section 2.0. This system used VAX 11/780 computers to provide the man-machine interfaces and an IBM 3081 to do the real numbercrunching work, especially for the simulation. The simulation is necessary in particular to do a realistic presentation of 3D design images. It is foreseen that larger computers had to be used in the future.

Also the CAD system which used a graphics station connected to a supercomputer was used for simulation, in particular for molecular dynamics and numerical analysis.

Another area where simulation plays a very important role is in VLSI design. Examples were studied where special multiprocessor systems were built in order to be able to do the simulation on the logical level. When simulation has to be done on the analogue level, the computational requirements become even more difficult to meet.

2.4 Database integration

Database support for the next generation CAD systems must provide combined models for geometry, product and production information. The already hierarchical structure of the geometric model, will have to be extended. This puts heavy demands on the database, which minimally has to be of a relational type. This is considered to be a next step towards knowledge engineering for CAD.
A good example of database integration is given by the system under design at the University of Tokyo, as described in section 2.0. The objective of the design of this system is an integrated and flexible automation system based on CAD and using additional assembly data. All this data will be stored in a database that is of a relational type. Later knowledge based systems are foreseen.
Because the process of engineering is well defined, they believe that it is very attractive to store engineering data into such a knowledge base.

Another example was the CAD/CAM system for an FMS line that allowed for the production of small batches of product variants. The design of the assembly and manufacturing was done on an in-house developed CAD system. Also all subsystems of a production line (software, electronics control, robot systems, etc.) were designed in-house. The CAD/CAM system made extensive use of a database for the integration of the various system components.

3.0 Computer Aided Manufacturing

Two main areas can be distinguished in which computer automation will play an important role in the near future. The first one is machining, the second one is assembly. The state of the art at the moment is such that machining lines are quite often already completely automated, whereas assembly lines are in the process of becoming being automated. The examples that will be given here are obtained from car industries, car component industries and machine tool industries.
As a first example a short description will be given of a car manufacturing production line. The example will describe
1. an engine machining line
2. an engine assembly line
3. a car assembly line

1. The machining line operated completely automatic. Manual interaction only takes place when tools had to be replaced or when a machine went wrong. The status of each machine was totalized on a large monitoring panel at a central position visible from everywhere.
   Machines were connected by conveyer belts, but worked independently. After every machining operation, the results were checked by sensing. All machines operated in sequence. Machine failure handling strategy was to do fast repair on the spot.

2. Most of the assembly was still done manually. The line was set up such that the beginning could be done greatly automatic, thereafter it operated only partly automated, whereas the end as well as the engine inspection were done manually.
   The engines were transported by trucks and mounted into a car within an hour. More then 85% of the factory manpower was being used for assembly.

3. A computerized system issued the orders for various parts to be supplied from distribution points. Upon entering a carbody in the assembly line a worksheet was generated. This information was sent from the central control room to the line. Full assembly required 20 hours.
   Every assembly procedure was completed by a test. Whenever a test failed, the car was marked and at the end put into a repair line. At the end of the assembly line, visual inspection and functional tests were carried out.

3.1 Flexible Manufacturing Systems (FMS)

The requirements to produce different variants of the same type of products demands for more flexibility in production lines. This argument is valid for machining, as well as for assembly. As machining can already be performed completely automatic in many cases, the assembly is the most critical part. To be able to produce a wide range of product variants, flexible manufacturing techniques were introduced. Flexible manufacturing can range from designing a completely robotized manufacturing line, via robotized manufacturing islands into a not completely automatic manufacturing line.

An example of the first case is the FMS line for the production of small batches of product variants of car meters. This highly flexible assembly line exhibited a number of interesting properties.
First of all there was a control flow device, controlling each station in the line and processing its feed-back.
Secondly an operator could define on site an arbitrary number of batches of different meters to be produced in sequence. The batches were seperated by a dummy component that was detected by every assembly station. Upon detecting a dummy, the local control sent a signal to the line controller and then received the information concerning the task for the next batch. On each station each assembly operation was immediately followed by a testing operation. The assembly line completed a part every 0.9 seconds. The average production time was reduced by a factor three. Subsequently a video tape showed a number of equally advaced assembly lines. For instance, there was a radiator line that could produce 64 different models.

It is possible to equip assembly cells with adaptive control capabilities, for instance, for callibration of meters and adjustment of a relais switching gap. On the average 300 different batches are produced per day.

A second example is an FMS line for the machining tool industry. An FMS line was studied that could produce three types of machine tools:

1. a NC Lathe
2. a NC Cutter
3. a NC Grinder

The FMS cells can be equiped with part handling robots and pallet based input/output. All the machines make extensive use of sensors for tool checking, as well as for the control of products. They can also produce the FMS machines for the FMS line.
Every machine is fully controlled by a home-made built-in controller. For each machine tool the man-machine interface is adapted to the machine task. A Computer Vision/CAD system is used, and an in-house developed die system, as well as an NC control system.
A central computer connected via a fiber optic link with 7 machine controllers, controls the FMS line including an unmanned carrier. Using the built in sensors, 3D automatic measurements are possible, which can give feedback. Design of parts of FMS cells on customer specifications (40 parts with 3 machines with an onboard robot). is also possible. Several unmanned production lines are operational. Machining in these lines is done with an accuracy of less then 20 micrometers. The inspection data is the feed-back into the control units capable of adaptive control, when the measurements are outside the tolerances.

3.1.1 CAE for flexible manufacturing

The use of computers to assist in designing for Flexible Manufacturing Systems (FMS) is still very limited. Development is still in a preparative stage of rationalisation both for product components and tools. This rationalisation even without integration has already resulted in remarkable improvements.
As an example the flexible manufacturing line discussed in section 3.1 has the possibility to assemble 288 product variants of meters. The design for the part assembly and manufacturing system was done on an in-house developed CAD system. Also the subsystems such as software, electronics control and robot systems, were a result of in-house developments. These developments only took place after carefull rationalization of the total production activity. In such a way it became possible to define company standards.

3.1.2 Component strategy for machining and assembly

The introduction of FMS has resulted into a new philosophy in CAE. Standardization is introduced to be able to design the same variety of products with far fewer components. The FMS line, as was discussed in section 3.1 was able to produce 288 different variants of the same meter types. This could only be done by a far reaching rationalization of the product components. By doing so the company was able to define their own standards for components. These standards were chosen in such a way that the variety of components between products is low, whereas the product varieties can still be extremely high.

The reduction of the amount of components and their longer lifetimes make it economical to have component databases. Setting up a new product line can be done faster now. The integration of a component database with the CAD/CAM systems would result in a further reduction of lead-time.
These strategies are currently being implemented in machining and assembly of components.

A CAD/CAM system was studied that was able to produce FMS planning based on custom specifications of their products. It was foreseen that this type of facility would be extended in the future. Now the system is able to produce planning charts; in a later stage it will be connected to the FMS line. As a first step to reach this goal the CAD system is using simulation in order to study the various mechanical properties of their FMS line.

3.1.3 Tool strategy for machining

Similar to the component strategy, a machining tool strategy has been developed which attempts to minimize the number of tools without reducing machining flexibility. This can also lead to tool databases and computer aided production engineering.
The rationalisation of the production process that leads to standards in both components and tools, puts heavy constraints on the design process. The CAD system should provide powerfull assistance to compensate for the loss in design flexibility. The introduction of such a CAD system requires extensive training and initial tolerance of the designer.

A completely automatic factory was studied that could produce 200 flexible machining cells per month. The factory had two separated lines. The box line producing small parts, the other line producing big parts. On average a tool drum is changed once a week. In this factory all production could be done with 60 standardized tools. Every tool drum has a maximum capacity of 80 tools. A central computer is used controlling the whole line, the machine controllers are connected via RS232 connections.
In this factory machining was done practically without manual tasks. Parts handling was taken care of by unmanned conveyers.
The first of these factory lines applied 55 tools to machine 60 different products. A newer line had an even higher flexibility. The machining cells were able to reproduce themselves.

3.1.4 Design strategy for FMS cells

The most important goal in FMS is to reduce the lead-time. Lead-time is almost entirely determined by flexibility. Flexible machining and manufacturing can be realized by using programmable FMS cells of which only a few basic types are needed.
Programming of FMS cells is currently done locally on the shop floor, using computer generated NC programs. Studies are in prograss to automate FMS cells properly. This would allow for adding new production processes without bringing production to a halt.
The CAD/CAM for FMS is necessary, but by itself does not reduce lead time, this is because special tools and fixtures still have to be prepared.

Automation of NC-programming does not change this situation. The only solution is total rationalisation of CAM processes, which results in standardisation of tools and fixtures. Next these standards have to be input to the CAD system to obtain integration. An essential condition for using standards in CAM is that they have a user friendly interface especially for the designer. For instance, there is a need for an enhanced man-machine interface for interactive NC-programming, something like APT is considered useless in such an environment.

Because of the standardization the number of machining tools in the example of section 4.3 went down from 300 to 72. The absolute maximum number of tools a designer is allowed to use is 80, equivalent to the maximum number a machine drum can hold.

4.0 CAM with FMS systems

Current technology allows for entirely automated materials handling eg. unmanned cars, conveyor belts. Only simple materials handling is possible without manual support. The material handling by the robot can be simplified considerably by the application of standard pallets. Assembly lines for small components also use materials in well defined positions. In these situations material handling can be taken care of by simple teach-in robots.

4.1 Current use of robots

Teach-in robots are extensively used. Manufacturers mainly apply their own defined robots. Robots are either used as free standing material handlers or as on board robots in FMS cells. Currently most robots are not able to do material handling for assembly which is more complicated. However, this situation appears to be changing rapidly.

4.2 Manufacturing process control

The programming and control in machining and assembly lines is done locally, however switching of functions is controlled centrally. Consequently the monitoring of the status of each manufacturing cell is centralized. The design of the manufacturing process is done manually. For the future the integration of CAD and CAM systems by the application of networking technology is foreseen.

4.3 Sensing and quality control

For both machining and assembly the basic principle is to immediately check every function after it is being performed. Sensing is introduced in a guaranteed simple environment. The sensor data can be used either for feedback or rejection. The advantage of this approach is that quality control of the final product can be limited. This is another example of stepwise improvement.

Sensor systems are also used in cars for the control of

1 the car engine
2 the car transmission system
3 the brakes

The engine control system applies sensors for: fuel injection, air flow, spark control, exhaust control, compression etc.
All the sensor data is input to a microprocessor which processes this data and provides real-time feed-back to three actuators e.g. fuel control, air input and motor timing control.
Sensors are also used in warning systems among others for headlights, airconditioning etc. In the future, cars are foreseen, that have more advanced sensor systems with wider applicability.

5.0 Future trends in CIM

There is a tendency for further integration of all CIM modules. The software systems required for this next step are very complex. Integration is considered necessary to guarantee future competetiveness. The simplified technology resulting from standardization makes technology transfer to cheap labor countries easy. To keep ahead, further integration is the answer.

Total integration requires many steps which eventually can lead to knowledge engineering for production design of totally unmanned factories.
For these ultimate systems Fifth Generation (FG) computers or equivalent resources are considered necessary. The ICOT FGS project is considering CAD/CAM as one of the first areas for the application of their knowledge based systems.
These totally integrated systems must be programmable by higher level languages.
One of the outcomes of the FGS project could be such languages.

5.1 Future robots

A new generation of robots is in their experimental stage. They make extensive use of sensors in particular image sensors. They are able to recognize a limited amount of objects, in casu the standardized parts. Their major application will be in assembly and quality control. These robots will require a decentralized powerful processing support. Each joint as well as the viewing system will require its own local processor which autonomously contributes to the overall task. Communication with the central control processor is on the functional level.
So the next generation of robots will still be a compromise between a simplified environment, and preprogrammed tasks on the one hand and built in intelligence in the other.
Programming may still be a combination of teach-in and program control of robot functions.

5.1.1 Robot program

As an illustration of the various possibilities of robots in the future, the main characteristics of the MITI robot program will be discussed.
Part of this program was directed mainly towards low-level research, e.g. non-intelligent robots, but also advanced systems are under design, capable of intelligent actions.

- Mechanics of robots (vehicle, nursing robot, six footer)
- Robots with vision system for movement control
- Pick and place robot with visual perception
- Intelligent vehicle (obstacle finder)
- Guided dog robot (obstacle finder)

When robots were designed to sophisticate the mechanical movements, the recognition task was reduced to a minimum. For instance: recognition by means of external beacon systems like marks in a road.
One additional topic was the development of a redundant and fault tolerant microprocessor (6800) system to control various robot functions.
Work is done on arms, legs and viewing systems. A robot leg system with three degrees of freedom was demonstrated that had feed-back on hip and knee level. The computer system applied used a separate processor to control each joint. Overall control was done by a supervisor computer controlling the others.
A robot arm with six degrees of freedom was used in combination with an image processing system to illustrate hand-eye movements and coordination. A rope was positioned above a needle with a large opening. At the moment the rope was above the needle opening the vision system took over and was able to let the arm put the rope through the hole and making a knot in it. All the time the vision system was used to measure the rope in space.
The vision system was also able to track a moving object which was demonstrated using various toys. A wrist was designed with six degrees of freedom and feed-back at each joint. The first aluminium based construction had now been replaced by a carbon fiber enforced vetronite version resulting in an extremely light construction. Control of the joints was done by separate computers.

5.1.2 Image processing

Three projects were illustrated:

general image processing facility
laser based scene analyser
glossy objects recognition

<u>General image processing facility</u>

The hardware consists of a 256x256x4 frame memory as input, and 256x256x4 image memory as output. Input and output memories as well as a special image processing computer, which performs fast image processing operations such as histogramming and region filtering are connected to the same bus.
This system is used for scene analysis based on gradients in light distribution.

A demonstration of an application of this system was given where position and movement direction of people walking through a corridor were detected. First the contour of the people were found, thereafter a minor image in the shining floor was defined and the footprints of the person were detected. These were used to define the direction of the movement.

Laser based scene analyser

The device consists of a laser which makes a line source which is reflected by a shining object into a 256x256x4 bits CCD camera. The whole set can move in X and Y to scan a presented object.
Each point of the object is measured in triangulation, doing so flat and curved surfaces can be detected and measured.

Glossy object recognition

A glossy object was illuminated from a number of polarized light sources and scanned by a fixed camera. Off-line it was tried to reconstruct the glossy object using graph matching techniques.

5.1.3 General

Special purpose languages are under design for typical robot functions such as manipulation, vision and spatial control. Macro's specifying special tasks could also be defined by means of a master manipulator.
A national language translation system could make the translation from Japanese into English. It also gave a translation into logic expressions with the purpose to apply this in logic programming.

In the processing, proximity sensor information is considered to be low level because no use is made of external knowledge such as information collected by other sensors or data already available in the database.
This justifies the strategy of local processing of all of the sensor data coming from a particular robot function. Such a robot function is performing one task at a time. Combining tasks (such as obstacle finding and avoidance) is not considered feasible.

In the detection of moving objects the recognition of a global environment based on feature extraction is the most important thing. Local proximity sensors can give more detailed information.
For robot function processing, e.g. for vision, special purpose hardware gives significant more improvement then using a higher level language (Lisp) on special purpose machines.

Conclusions: the major improvements can be found in low level fast algorithms. Prolog and Lisp are not suitable to program low level vision algorithms.

5.1.4 Robots for critical work (RCW)

This national program contains the following subjects

Locomotion M
Manipulation M,E
Sensoring E
Actuation E
Hand-eye coordination M,E
Advanced teleoperation (global commands) E
Tele assistance (commands to operator) M
Total systems M,E

M = Mechanical engineering
E = Electrical engineering

It is expected that this program will bring about both a basic understanding of intelligent robots and a considerable increase in flexibility of robots, and therefore a considerable increase in applications of industrial robots.

5.2 Future CAD/CAM systems

The requirement for more intelligent assistance for the designer will require more dedicated processing power combined with multifunctional data representations. On both subjects concentrated R&D efforts take place. These may well lead to a new generation of original highly effective CAD/CAM systems.
The integration will require two new types of man machine interfaces. One is a highly powered CAD/CAM interface for the design work as mentioned above, the other one is associated with overall CIM systems monitoring.
Many of the new applications will use VLSI techniques. VLSI design will become a common tool for a CIM systems designer and implementor.
A second strategy for more intelligent design assistance is the use of knowledge bases. CAD applications are one of the first target applications for knowledge bases under development. Engineering, being a well defined discipline, has relatively few problems associated with organizing the required information.

5.3 Future Flexible Manufacturing Systems

In the FMS experimental factory project, three governmental institutes are collaborating with 20 private industries. Purpose of this research is to create a fully automated factory which, when in operation, does not need any human interference. However, programming of the factory is done completely manual.
The factory applies laser technology for metal surface treatment. The laser beam is generated centrally and can be led to (and integrated with) every machining cell.
The experiment aims to integrate various machines designed and produced by e.g. different participating manufacturers. Integration with CAD systems is not considered.
Process planning is performed manually. A processing strategy will be developed that considers the integration of the various machines fundamental and a prerequisite for any further integration such as with CAD.