

SRI International



ENGINEERING DESIGN VIEWED AS AN ACTIVITY IN ARTIFICIAL INTELLIGENCE

Technical Note 167

July 24, 1978

By: Kenneth Preiss
on sabbatical leave at Poulter
Laboratory, SRI International from
Department of Mechanical Engineering
Ben Gurion University of the Negev
Beer Sheva, Israel

ABSTRACT

The process of engineering design, and its interaction with society and with production methods, has been studied for the past few decades. The process of problem solving has been studied in artificial intelligence research for the past few decades. This paper describes the engineering design process from several viewpoints, and then relates this process to procedures that can be implemented using computers. As a result, aspects of the design process are clarified, and the directions in which computers can make increased contributions to this process are highlighted.

CONTENTS

ABSTRACT	ii
1	INTRODUCTION	1
2	ENGINEERING DESIGN: THE CONCEPTUAL VIEWPOINT	3
3	ENGINEERING DESIGN: THE PROJECT VIEWPOINT	6
	3.1. Feasibility Study	6
	3.2. General Design	7
	3.3. Detailed Design and Prototype Construction	7
4	ENGINEERING DESIGN: THE ADMINISTRATIVE VIEWPOINT	9
5	CONSTRAINTS UPON THE DESIGN PROCESS	10
	5.1. Relations between Products in a Modern Industrial Society.	10
	5.2. Time Dependence	12
	5.3. People Dependence	14
6	CONSTRAINTS TO BE IMPOSED ON A MODEL OF THE DESIGN PROCESS	15
7	ENGINEERING DESIGN FROM THE MACHINE INTELLIGENCE VIEWPOINT	16
	7.1. The Information Network	16
	7.2. Instantiation by Constraint Propagation	22
	7.3. Analysis Formulae and Inference Rules as Domain Switches	26
	7.4. Constraint Decoupling Frames	27
8	FEASIBILITY STUDY: ESTABLISHING A DESIGN CONCEPT	29
	8.1. New Design and Redesign	29
	8.2. The Process of Designing Data Frames	29
9	DESIGN DURING THE GELLED PHASE	33
10	ENGINEERING DESIGN IN THE FRAMEWORK OF THE MODEL	35

11	SOME EXISTING WORK IN COMPUTER-AIDED DESIGN REVIEWED IN THE LIGHT OF THE MODEL	38
12	RESEARCH IN ARTIFICIAL INTELLIGENCE IN THE LIGHT OF THE MODEL	40
13	CONCLUSIONS	43
	REFERENCES	44

1 INTRODUCTION

Computers are being integrated into the technological design process, just as they are being integrated into banking practice, airline reservations, and many other activities. Up to the present, computers have served as tools used by designers in methods dictated by the paper-based information systems to which humanity has become accustomed. Computers perform design calculations or analyses, and produce documentation such as drawings, schedules, specifications, and tape to control production machines. However, as more and more technical information is stored in computer memories, and as the communication network between computers develops, different working methods, adapted to electronics-based rather than paper-based information systems, may develop.

This paper explores the process of design, and then, using some concepts published in the last few years, mostly in the field of artificial intelligence, proposes a model of the design process that can be implemented by computer. The paper does not give a comprehensive survey of past research and writing in the subject of problem solving in engineering design; concepts from that literature are included and these are referenced whenever appropriate. Furthermore, the ideas expressed here have not yet been used to create a working design program. The examples used as illustrations of various concepts are therefore only partial.

The word design means different things to different people. Here it is used for the general process whereby an idea in someone's mind is developed, refined, and elaborated into the detailed instructions for manufacturing a physical product. Furthermore, the product considered will be produced by many men and machines. The design and production of a statue by an artist, for instance, does not fall into the category of activities considered.

The concept of design is illustrated in Figure 1, from Dixon (1966). This figure shows that the activities of scientific discovery, analysis, and design may all deal with the same entities, but that they are different activities. A single person may on different occasions perform acts of scientific discovery, analysis, or design, but the activities are nevertheless different.

2 ENGINEERING DESIGN: THE CONCEPTUAL VIEWPOINT

Figure 2 shows an abstraction of the design process; this diagram is also from Dixon (1966) with some modifications.

It is seen that the design process starts with a definition of a need, which can be satisfied by the product. This need may take many people and much time to define, or may be specified by one man in a few minutes. The definition is then gelled by formation of a general concept of the solution required, which gives rise to a specification for the product. After engineering analysis a solution is specified, and then production commences.

In current computer-aided design activity, computers contribute to the computation phase in engineering analysis and to producing the documents that specify the design solution, such as drawings and schedules. Although the prior decisions, in which the design concept is formed and the computational models are selected, are important, computers have as yet made little impact in these areas.

Design is not a simple serial process. Each step has to take account of each of the other steps. For instance, product specification has to take account of the following sets of constraints:

- * Constraints due to the user of the product -- The properties of the product have to be constrained so that it will in fact be used by potential users. If the product being designed is a new model of an existing product, it may be easy to identify the constraints with marketing surveys. But if the product is new, it may be very difficult to identify the user constraints. This consideration is as important in a centrally planned economy as in a private enterprise economy, because production of a product that does not give economic benefit at least equal to the effort in producing it wastes resource, whatever the economic and social system.
- * Constraints due to the design process -- The product should not be specified in such a way as to require design

analyses or procedures that are not available. For instance, if the design specification requires very precisely defined response to physical vibration at different frequencies, then the design process must include the ability to analyze the design analytically or experimentally for those responses.

- * Constraints due to the production process -- The product must be defined so that the men and machines available for production can produce it.
- * Constraints due to the testability of the product -- The product must be defined and produced so that a user can test it before using it.
- * Constraints due to distribution -- Distribution systems may be different in different kinds of economies, but in any case the product must be designed so that it can be distributed. This may require constraints on size, shape, and weight, and may require design to withstand accelerations, temperatures, humidity, and changes over time.
- * Constraints due to maintenance -- the product must be designed to be maintainable. Typically, this requires modularity and standardization of components.
- * Constraints due to retirement -- This aspect of a product has been rather neglected in the past, with the result that difficulties are experienced when wanting to retire a product. Typical examples are the retirement ("junking") of automobiles, or the demolition of structures.

It would require an exceptionally perspicacious designer to foresee all the design conditions that would have to be considered as the design progresses. In Section 5 it will be argued that such foresight is not possible. As a result, the various phases of the design process interact with each other. A designer tries to foresee all the possible interactions, and tries to constrain the various stages of the design process so as to avoid far-reaching interactions. For instance, it is most undesirable to discover at the detailed engineering-analysis stage a factor that would require a different concept to satisfy the identified need; yet such may happen. A good designer, by experience and art, defines the design such that if constraint conflicts occur, they can be solved without far-reaching changes to the whole design. The words of Newell and Simon (1972), related to problem solving in general, are applicable here: "Determination of goals and operators, and evaluations of new states, seem the important controlling decisions."

Several textbooks on engineering design describe methods for enhancement of creative design. These include

- * Brainstorming -- The aim is to record as many ideas as possible, without weeding out bad ideas. Selection is made later, but in the brainstorming session one hopes to have people trigger each other to new ideas.
- * Empathy -- The designer tries to imagine he is the item being designed and tries to discuss the influences acting on the item.
- * Matrix Evaluation -- If both the problem and the solution have several domains, these are plotted in matrix form to ensure that no possible combination is forgotten. For instance, if the problem is to remove pollutants from water, the problem domain may be split up by type of pollutant, its concentration, required throughput, and possibly other variables, and the solution domain is split up by treatment methods. In some organizations, standard checklists are available for structural and mechanical designers.

3 ENGINEERING DESIGN: THE PROJECT VIEWPOINT

Although the process outlined in Section 2 accurately defines the design process, only small design efforts can be administered by that scheme. Any effort involving several or more people is organized as a project. Ostrofsky (1976) describes the design process in this way: a project is divided up into phases, which usually are feasibility study; general design; detailed design and prototype construction; and construction of a small production run, with design updates to account for quantity production methods.

3.1. Feasibility Study

The aim of a feasibility study is to find a feasible solution to satisfy the needs identified as the problem definition. At this stage, the problem is usually incompletely defined; further, the different solutions pose different questions to the problem definition, and these require new specifications.

Figure 3 shows the stages in a feasibility study. The end point of an engineering feasibility study is a definition of an engineering system. The major concepts are defined, but the general and detailed engineering design lies ahead. The feasibility study provides guidelines and constraints to the engineering designer. In the architectural world, the functional program by which the structure is to be designed is the output of a feasibility study. Such a program would give information about the functions to be fulfilled by the structure, but would not specify where the rooms and facilities are to be.

A feasibility study may be done by a large team with much time, or it may be done in a short time by an individual who engages the designer.

Executing a good feasibility study requires experience and art, so that the output specification (which is input to the designers)

- * be definite enough so that the design work will not waste resource by focussing on irrelevant questions
- * be general enough to allow the designers to find good solutions
- * avoid an interaction such that later design work will lead to a change in the specification to the designers, hence wasting resource for redesign.

3.2. General Design

At the stage of general design, all the major subsystems are identified. Requirements of different subsystems are analyzed to ensure that the subsystems will be compatible and not conflict with the system specification. Decisions are made to decouple potentially interacting subsystems. For instance, in a structure, the method of space heating or cooling will be chosen, but not designed in detail. However, a decision will be made as to what size openings are to be left for the heating or cooling fluid ducts or pipes, and the rules established for where to locate these. Later, however, the interior space is partitioned into rooms, and whatever ducts or pipes are chosen, the room and air-conditioning designs will not interact, unless the details of the decoupling decision were inadequate.

Typically at this stage, subsystem suppliers will be chosen, and orders for components requiring a long lead time will be placed.

3.3. Detailed Design and Prototype Construction

Detailed design and prototype construction usually progress simultaneously. As a result, wherever interaction between subsystems is possible, and one subsystem is designed before the other, interface decisions must be made to avoid interaction effects whereby a second subsystem design will necessitate changes to the first subsystem, already under construction. For example, in building construction, design would logically be done from roof down, so that when the

foundations are designed, the weight of the structure is known, and, from the load/deflection characteristics of the structural system, dynamic effects on the soil or rock may be estimated. However, buildings are usually constructed from foundation up before the detailed design is completed. The solution to this conflict is to design the foundations for estimated loads and effects and to define characteristics of the structure, which are typically allowed differential settlements between columns, and definitions of the stiffness or dynamic properties of the structure. The foundations are designed for these parameters, and the structural designer ensures that the structure does not impose excessive conditions on the foundations. The decoupling decision requires experience. If the conditions are insufficient, the foundations and structure will be incompatible; if the conditions are excessive, the foundations and structure will require excessive, and therefore wasted, resource.

Many such decoupling decisions are made in the course of a design. Where they cannot be made, the interaction between subsystems must be integrated in the design procedure.

One of the major reasons for building a prototype for serial production is to discover unforeseen interactions, or to get precise information on interactions that were foreseen.

Some projects terminate with the construction of the prototype. For instance, most structures are one-off and hence are prototypes, even though each prototype usually will contain subsystems that are products of mass production (see Section 5.1). In any case, even a one-off product must be completely documented, because if it is used it will have to be maintained and possibly modified in the future.

A one-off item made by one man and maintained only by him (if at all) needs no documentation, but is then not in the scope of this paper.

4 ENGINEERING DESIGN: THE ADMINISTRATIVE VIEWPOINT

From a purely administrative point of view, design is a paperwork process with various stages, each of which accepts input documents and emits output documents. The total system must be coordinated to provide the required output at preplanned time intervals, and the message protocol (i.e., drawing numbers, document formats, etc.) must be coordinated over the whole project. Although this aspect is mentioned here only briefly, and not discussed in detail, it is a point of view commonly held, with much systematic methodology and practice.

5 CONSTRAINTS UPON THE DESIGN PROCESS

5.1. Relations between Products in a Modern Industrial Society.

Figure 4 shows diagrammatically the essence of a system to produce a physical product. Raw material and subassemblies are transformed by a production process into a product. The production system is constrained by the laws of nature, which define the properties of the raw materials and subassemblies, and by the capabilities (pressure, temperatures, rate of throughput, etc.) of the production machines. The production process is further constrained by considerations of the physical safety of the production workers, and by the physical and mental limitations of all the people involved in the production process.

This view of a production transformation is true whatever the sophistication, the size, or the type of production system. In the case of road or structure construction, the production system includes the trucks that transport soil, concrete, or steel; the quarries or factories that produce the raw material; the scrapers, rollers, and earthmoving equipment, and much more; and all the people who coordinate and operate the equipment. Taken together, these items comprise a system that converts iron ore, clay, lime, water, asphalt, and other materials into a product such as a road, bridge, or building.

The view is valid also for a machine making a lower-level item, such as a concrete mixer, which transforms rock, sand, cement, and water into concrete. Note that the raw materials for the concrete mixer are the products of previous production facilities, such as a cement factory or rock crusher, and the product of the concrete mixer is the raw material for the next stage in the construction process.

A primitive or preindustrial production facility performed the identical process of transformation. Whether we consider the stone-age

craftsman who fashioned a tool from flint or a seventeenth-century locksmith, the view of production as a process that transforms input raw material and subassemblies into a product, within the constraints of the capability of the men and machines who comprise the process, is valid.

Although we all live in a society sustained by a modern production system, not everyone is always consciously aware of all the characteristics of such a system. In fact, it is only in the past few decades that a serious effort has been made to view the cycle of product design, production, use, and retirement as a single system, and this is still an ongoing topic in engineering-systems research. Some of the major characteristics of a modern production system are reviewed here.

In a preindustrial production system, a single man, the craftsman, made his tools, obtained the raw material, designed the product, and made, sold, and serviced it. Today these functions are split up among many people working for different companies or branches of an industrial complex, often at large distances from each other.

Today, every process requires the output of some other process as input raw material, as an input material or subassembly, or as a production machine.

5.1.1. Input to Production Systems

The most basic level of the production system uses natural material as input. These natural materials include air, water, food, minerals, etc. Any other production process uses the output of man-made production processes as input. Since the input to any process is the output of some other process, one may define a ratio by which the input to all production processes in an economy is expressed as the ratio of (output of man-made processes)/ (output of natural processes) in the total input material. The more industrialized an economy, the higher that ratio will be.

5.1.2. Output of Production Systems

The output of any production system should have a use -- that is, it should be input to some other process. This concept includes food for humans and animals, machines and subassemblies, and materials for agriculture and industry. In fact, many processes have output for which there is no use. Sometimes a use can be created, sometimes the output cannot be used and leads to pollution or to a disposal problem.

5.1.3. The Transformation System

Any transformation process in an industrial economy requires equipment produced by some other process. In other words, in an industrial economy, any production process requires the output of some production process to provide the components of the transformation system.

5.2. Time Dependence

The design process takes place over a long time. The time may be a few months for a small project, or many years. There are a number of constraining conditions that themselves change with time.

5.2.1. Changes in Requirements

If the product to be designed will have little use, or if it is part of a slowly changing domain, there will be little interest in it at design time, and no demand for changing the product specification. But if the product arouses interest, is new, or is part of a rapidly changing field, astute salesmen and managers will find further applications for it while it is being designed. These further applications may require changes to the product specification. The chief designer is then caught between the desirability of accepting specification changes, which will increase the use of the product, and the desirability of not accepting any change, thus keeping down time, cost, and the probability of design errors.

Often the properties of a product are determined by the properties of other products -- e.g., military missiles, the properties of which are affected by adversary missiles, or components of automobiles affected by changes in the total automobile design. There is then little choice but to accept a requirement change.

5.2.2. Changes of Component Properties

The design may include components made by a particular company, and these components may change due to technical changes such as model updates, or due to management changes, mergers, bankruptcy, etc., of the supplier. Even if these changes do not occur, lead time for supplies may become so long as to provoke a shift to another component.

5.2.3. Changes of Manufacturing Capability or Distribution Method

In an ideal world, the designer would know of an impending change in the technical properties of manufacturing or testing equipment, or distribution equipment, and would take these into account. However, the real world has unpredictable events due to strikes, illness, weather, and natural disaster, and so the interactions between management, production, and distribution often appear to an individual designer as unpredicted changes.

5.2.4. Time Rate of Condition Change as a Function of Size and Importance of a Project

The larger a project, the more people and time are involved in the design, the more components are used, and the higher the probability of a change during the design process.

Resources invested in a design show no return until the product begins to be used. As a result, design is often under pressure to minimize the design effort. This concept is valid in a Western commercial-type economy, but is also valid in a planned noncommercial economy, because the resources used for design of one product can always

be used on other projects; and the resource expended in design must eventually be repaid to the economy as a used and useful product.

The following comment by Kronman (1978), for the aircraft industry, is applicable generally in engineering design:

In addition to the large volumes of wiring data, it is a fact of life in the aircraft manufacturing business--verifiable by a long history--that there is a high rate of change in the design throughout the production program, especially in the beginning. These changes are made necessary for a number of reasons, including: changes in airline operating requirements, rulings from regulatory agencies (such as FAA), changes in subsystem designs by the system vendors and design errors. While each individual change may be quite small, the total change rate on a large project can be staggering.

To sum up, if a product is worth designing, it will in principle be designed in a time-varying environment of requirements. The design process must be able to cope with this time-dependent variation.

5.3. People Dependence

Just as conditions applicable to design subtasks vary with time, so do they vary with people. In an ideal world, people understand what they are meant to understand, do what they are intended to do, and are very creative up to a constrained border, but never cross the border. In fact, instructions are misunderstood, people do what they were not intended to do and do not do what they were intended to do, and their creative thoughts do not recognize administrative boundaries. Even if the conditions applied to the design subtasks were constant with time, they would vary as they are dealt with by different people. The design process should help designers be creative, but help managers find and resolve conflicting design actions of different designers.

6 CONSTRAINTS TO BE IMPOSED ON A MODEL OF THE DESIGN PROCESS

The model of the design process is to be implemented in a computer system. This implies that the model need be only as accurate a reflection of the real process as will suffice to be usefully represented in a computer-based design system; it need be no more accurate than that. This definition is not precise because no specification is given to the concept "useful." As different systems are tried over the years, and as research in man/machine systems progresses, a more precise definition of what is useful in different fields will doubtless evolve.

Constraints that should apply to the model are:

(1) The implemented system should enhance creativity, but enable designers to identify conflicting subsystems and resolve those conflicts.

(2) The system should advise the chief designer (or chief design team) of the implications of a proposed change in product requirement or in component properties.

(3) If a change mentioned in (2) above is to be adopted, different methods, or different optimal methods, for integration of the change should be found.

(4) The system should accept input from a designer in a way that includes understanding of the designer's desires, and the implications thereof. That is, the context and the inferences of the context must be understood.

(5) The system should advise the designer of the reasons for any decision made or question asked.

(6) The computer can obviously use data only from the bounded data base available to the processor. In principle, the designer's responsibility is absolute, and so he must be aware of domains of data and interactions about which the computing system is not aware. The model must therefore allow the designer to add information or to override the computer's decisions.

7 ENGINEERING DESIGN FROM THE MACHINE INTELLIGENCE VIEWPOINT

7.1. The Information Network

7.1.1. The Data Frame

The output of design is the definition of the product to be manufactured. This definition is an aggregation of information, given by drawings, parts lists, and specifications. The frame concept of Minsky seems appropriate for describing the organization of this data (Winston, 1977).

A frame is a remembered data structure for representing stereotyped information. Attached to the frame are several kinds of information, accessible via terminals. A frame includes the data needed to specify a subassembly or product at that level, but does not add details from lower levels. For instance, a data frame defining a pump to be used in, say, a building, would define all the characteristics needed by the designer who uses the pump, such as physical dimensions, weight, pressure and flow characteristics, physical layout of inlet and outlet apertures, holes to fasten the pump in place, and similar information. The frame would not include more detailed information unnecessary to the pump user, such as details of the nuts and bolts that hold the pump together. See Figure 5.

The frame could include semantic or conceptual information, which today is held in the mind of the designer, such as safety rating, reliability of the product, reputation of the manufacturer, and the like.

Evidence from psychological research (Newell and Simon, 1972) shows that much human problem solving is serial -- that is, only one step is processed at a time; that the information processing procedure uses data held in short-term memory; and that about seven chunks or

aggregations of - data can be held at one time in the short-term memory. These details seem applicable to engineering design.

The frame structure certainly contains much nonnumeric data. Usable theories of how these data are analyzed during problem solving are now becoming clear. It is as yet not clear just which data are used by designers in different fields, and how the data interrelates. With time, the data will be elucidated for different fields, and analysis of these nonnumeric data by the computer for the designer will be a valuable tool.

In the design process, one may consider a frame to be a tabular array with titles but with no values initially entered. As design progresses, values are assigned to places in the tabular array. This is called "binding of a variable" or "instantiation of a variable."

All designs use components and materials made available by other design and production processes. Since such items are used as they are, the frame representing each such item is already fully instantiated and cannot be changed by the design in progress.

The meaning of the word "variable" should be elucidated. A variable can be considered as a label or name to which a set of possible bindings belongs; one of those must be chosen and bound to the label.

For example, consider a bolt on the fan of the cooling system of an automobile engine. Labels will be bound, to illustrate the concept.

For the label <vehicle>, possible bindings are <automobile>, <pickup>, <truck>, <ambulance>, <fire-engine>. The binding <automobile> is chosen.

The data frames that form subsystems of <automobile> are <chassis>, <traction-system>, <suspension-system>, <steering-system>, <interior>. The binding <traction-system> is chosen.

The data frames that form subsystems of <traction-system> are <engine>, <gear-box>, <clutch>, <drive-shaft>, <braking-system>, <differential>, <propeller-shafts>. The binding <engine> is chosen.

The subsystems of <engine> are <cooling-system>, <lubrication-system>, <fuel-system>, <valve-system>, <piston-and-cylinder-system>, <ignition-system>, <crankshaft-and-bearing-system>, <flywheel>. The binding <cooling-system> is chosen.

The subsystems of <cooling-system> are <fan>, <radiator>, <cooling-fluid-circulation-channels>, <pump>, <thermostat>, <temperature-sensors>. The binding <fan> is chosen.

The subsystems of <fan> are <blades>, <bolts>, <shaft>, <bearings>. The binding <bolts> is the binding required for this example.

The path of the bindings through the frames is <vehicle>_<automobile>_<traction-system>_<engine>_<cooling-system>_<fan>_<bolt>.

Variables are thus seen to be instantiated from the general to detailed levels in the frame system. The data frames can be thought of as arranged in a tree structure, with the frame for <vehicle> as the root or goal node. Instantiation stops when a frame is reached that represents a part not requiring design, or a part made on a press or machine tool and then used. Such frames cannot be further instantiated in the design process and are said to be at a ground state. They are tip or leaf nodes of the tree. Binding proceeds, therefore, until a ground state is reached. In the above example, if the designer had to use, for whatever reason, an engine Model E of manufacturer M, then the binding <engine> would have been a ground-state binding and no further instantiation would have been possible.

From a strictly logical viewpoint, it is not necessary to aggregate the data into frames. All the information could be kept as separate entities in an organized list of independent facts, rather than on drawings and specifications. It is readily seen that searching for data would then take much time and effort, and is the reason the data is organized into frames.

When designing with a computer-based information system, the same consideration applies. Data can be stored in the computer as purely independent facts, called upon as needed. Such a relational data base (Martin 1977) would permit all the logical activity of design, but

the access time required for getting the data together for any one calculation or instantiation would be too long. An organization is needed that makes available at any one time the data needed for instantiation, and only that data. Hence the attractiveness of the frame concept.

The structure of the data frame, or, in other words, the names of variables to be placed in a frame (and the meanings of those names) must be chosen not only with the considerations of data base design in mind, but also with the considerations of control of a program taken into account. Therefore, to instantiate data in a frame network in hierarchic fashion, it is necessary that each frame be easily related to its daughter frames: each variable at any level L must be a subset of some variable in a parent frame at level L-1 (a subset includes, of course, the case of a variable being copied from level L-1 to level L). The set of relations between variables in the hierarchic direction must be known for each frame either as a part of the frame or as a set of rules held separately but attached to the frame when needed.

Because of this hierarchic definition, a constraint applied to a variable at level L must be valid at level L+1 because each variable at that level is a subset of a variable at level L. Furthermore, this allows inclusion of bottom-up considerations in top down control.

For instance, a designer may be designing a machine that includes, say, a fan, and let us assume that a particular fan type F1 of a particular manufacturer has to be used, not for any technical reason, but because that fan has been bought at a good price, or because the fan manufacturer is a sister company of the machine manufacturer. This appears as a constraint on the goal node. Among the variables on the goal node will be one that has the variable "FAN" in one of its subsets; at some level "FAN" becomes the identifying name of a frame. The constraint is carried down from the goal frame to the frame for the fan:

```
IF (FRAMENAME = FAN) THEN ((MANUFACTURER = M).AND.  
                           (TYPE = F1))
```

(all other data for the fan are available as ground information).

7.1.2. Drawings and Specifications as Data Frames

Frames of data are organized in two general ways. The first is hierarchically according to the hierarchic relations between the product and its subsystems, as they are manufactured. The common practice is to have a general layout or assembly drawing, together with specification documents. These refer to other drawings and documents for details of component subsystems, and so on, until an already instantiated or ground frame is arrived at. The ground frame would refer to a product incorporated into the design, and therefore completely specified, such as "motor, catalogue number N1, manufacturer M, defined by national (or industry, or company, or military) specification S."

Each frame in the subsystem hierarchy may hold data from different domains, where a domain may be length, mass, color, electrical properties, etc.

The relation between data in one domain but in different frames may be shown by a domain-specific frame. For instance, an electronic instrument would have both kinds of data frames. The first is a hierarchy of physical subsystem drawings, starting with one complete layout, for which considerations of all domains such as electronic, waterproofing, shock (packaging), aesthetics, and human engineering are taken into account, then on down to detailed drawings, which at the lowest level show component layout on a board. An example is shown in Figure 6a. In addition, there would be a circuit diagram, which is a domain-specific frame, showing how the electrical components operate as a system, wherever the components may be. This domain-specific data frame may also be arranged hierarchically, as shown in Figure 6b.

Similarly, in a building, general drawings (or frames of data) show, among other things, all the water pipes and connections, representing them where they are to be built, but there will also be domain-specific drawings (or frames of data) showing the pipes and connections as a water-supply system.

Note that on a drawing, much information is compressed into a few words in two parts of the drawing:

- * In the notes on the drawing, an annotation such as "Part A to conform to standard XYZ" can imply a large number of constraints on part A.
- * The title of the drawing contains much information readily understood by a human. Consider, for example, the title of a drawing for the mechanical part of electrical equipment called "connector." The drawing may include all the dimensions and tolerances, and names of materials used. The drawing does not usually give information clear to a human on the purpose of the connector, which is to provide a demountable connection for electrical conductors. This implies requirements for the connector not explicitly stated anywhere, such as easy access for moving the connector on and off; for intelligent assistance to a designer by a computer, the knowledge that handling space is needed, and what that space must be, must be in the computer.

The data frame of an item is therefore made up of

- * Information usually shown on drawings
- * Specifications
- * Meanings of words understood by designers in the field, and perhaps defined in dictionaries or technical specifications.

Corresponding to the hierarchic frame structure, a drawing contains pointers to the more detailed level of drawings and documents referred to by it. Much of the data about the subunits does not appear explicitly, but is included in the designer's understanding of the notes and titles on the drawings.

There is no one arrangement of frames to suit all designers or to suit the same designer at different times. The first task in design is to select the arrangement of frames suited to the task. This is considered in Section 8 and is analogous to the work up to establishing a concept in Figure 2 of Section 2. After the concept is established, the definition of the frame structure is made during the general design phase (Section 3), and then the frames are instantiated until at the end of the detailed design phase all the frames, to the most detailed level of the hierarchy, are instantiated.

In this paper, the phase of defining the frames is called the phase of feasibility study (Section 8). Instantiation of variables in a system of defined frames is called the gelled phase and is discussed in Section 9.

7.2. Instantiation by Constraint Propagation

7.2.1. Binary Constraint Relations

It has been established that design can be considered as the process of instantiation of variables in a hierarchic network of data frames. There is evidence (Henrion, 1978) that designers evaluate designs by evaluating binary constraint relations, initially searching for conflicting constraint conditions. Simon (1969) states that many design problems are "nearly decomposable hierachic problems." The meaning of this is presumably that an ideal, or decomposed, problem would be strictly hierarchic, but that constraint relations do occur between branches of the tree. If the between-branch relations were many compared to the hierachic relations, then the problem could not be considered hierarchic. However, the hierarchic structure seems to form a reasonable basis for the model.

7.2.2. Inference and Analysis as Constraint Relations

Instantiation of data in each frame is made by a designer by judgment and by calculation. Both these processes can be considered as propagation of constraint relations. For instance, a steel bridge designer may decide not to use a section thinner than half an inch to take load, because over the years corrosion can occur; a concrete designer may put extra steel in a job where he is not confident of accurate workmanship. An electronic designer may use a component because of considerations of reliability or reputation of the supplier.

These examples of propagation of constraints from domains which are not usually expressed numerically (corrosion and environment, quality of workmanship, reputation of the supplier) are examples of propagation by inference. Consider the above examples. For the case of

the steel bridge designer, corrosion in the outdoor environment is a domain he considered in the data frame for the bridge.

Instantiation of the value of thickness of a steel plate includes the following steps, some of which are illustrated in Figures 7a, b, c, and d.

The variable "environment" is instantiated to "seaside"

The terminal to environment is connected to a constraint relation, so that

(environment = seaside) implies (steel thickness not less than 0.5 inch)

or (environment = seaside) \rightarrow ($t \geq 0.5$ inch)

or (if (environment = seaside) then ($t \geq 0.5$ inch)).

In instantiating the value of the thickness of steel in a part of a structure, several constraints could apply. For instance, the thickness t of steel in the flange of an I-beam subjected to pure bending is given by

$$t \geq M/bhf_a, \quad (1)$$

where M is the bending moment, h is the beam depth, b is the flange width, and f_a is allowed stress. The bending moment M would be derived from knowledge of loads and geometrical layout of the bridge.

The girder depth may be derived from a number of independent considerations:

(1) Some functional reason, such as constraints on the road level above and clearance below, may imply an upper limit on h .

(2) Some functional reason, such as required room for connections, may imply a lower limit to h .

(3) If the web thickness is limited, then the maximum shear force in the section will give a lower limit to h .

(4) For equation (1) to be valid, $t < k_1 h$, where k is a constant (in practice it is derived from relations, usually functions of moments of inertia of the section and length dimensions, and is approximately 0.1).

The value of b would include considerations such as

(1) A relation exists between t and b , deriving either from a building code* or from aesthetics, such that

$$t > K_2 b,$$

where K_2 is a constant. (K_2 would be about 0.05 in practice.)

(2) A functional reason such as space available may give an upper limit to b .

(3) A functional reason such as required room for connections may give a lower limit to b .

(4) Allowed stress is a material property that is obtained by inference:

if ((material = mild steel) then ($f_a = 0.6 f_y$))

or, in different notation,

if (material = mild-steel) \rightarrow ($f_a = 0.6 f_y$)

or,

if ((material = mild-steel).and.($f_y = 33000 \text{ lb/in}$)² .

then ($f_a = 20000 \text{ lb/in}$)² .

By arranging a terminal for each domain on the data frame, and by propagating the constraints relevant to each domain, one can find the value of steel thickness. Inferences and computations are both constraint relations. Note that the propagation can lead to an insoluble problem, if two or more constraints conflict.

An analysis formula such as $t \geq M/bhf_a$ can be considered as an inference rule, relating right-side or antecedent bindings to left-side or consequent bindings:

```
if((flange-depth = h).and.(flange-width = b)
   .and.(bending-moment = M)
   .and.(allowed-normal-stress-in-bending =  $f_a$ ))
   then (flange-thickness  $\geq$  t).
```

Instead of a formula, a computer code, such as a finite element program, may be available. The code can be considered as an operator, which uses the right side or input data or antecedent bindings to produce left-side or output or consequent bindings.

* These and a considerable number of other rules governing the dimensions of steel beams may be found in standard reference books such as the AISC Manual of Steel Construction

The process of engineering analysis can thus be considered equivalent to applying inference rules.

Consider the example of the steel in concrete. The designer may use rules such as (typically)

```
If ((name-of-frame = concrete-slab).and.(workmanship = poor))
  then ((diameter-of-longitudinal-reinforcing-bars > 0.5 inch)
        .and.(bar-spacing < 8 inch).and.(provide 50% extra steel
        at slab corner, for 1/5 of slab width)).
```

An inference rule as shown above is valid and can be implemented, provided the data frame allows for a variable "workmanship."

An analogous example can be shown for the electronic equipment designer referred to above, provided the data frame permits instantiations of the variables "reliability" and "supplier-reputation."

7.2.3. Analysis and Iteration

The process of analysis and iteration is commonly used. In other words, assume a configuration, compute some values, change the configuration, and so on, until a satisfactory solution is obtained. This is a case of repeated trial instantiation until constraints are satisfied. To revert to the example of the steel girder, other constraints may be added. For instance, for reasons of economy, it may be desired that material be efficiently used, so a typical constraint may be

$$f_a \geq K_3 f_a,$$

where K_3 is a constant equal to, say, 0.9. Or, it may be specified that for the beam flange, steel of standard thicknesses and widths must be used. The relevant constraints would then be

<use a pair of values t and b such that the pair is an element in the set of (t,b) pairs available, and f_a is a maximum but not exceeding f_a >.

Stated otherwise, analysis and iteration is the same as applying inference rules and testing that the bindings obtained are acceptable, according to the constraints that apply to them.

7.2.4. Optimization

Analyze and iterate can be considered as a process of optimization; in the example shown, the stress f is optimized within the space bounded by all the constraints mentioned. Optimization can be taken one stage further, where some criterion, function G , is stated as a function of n variables x_i :

$$G = F(x_1, x_2, \dots, x_n),$$

and values of x_1, \dots, x_n are found to minimize or maximize G , within the limits of constraints on the x_i . Optimization can be used with safety in design only when all the variables x and the function F are recognized, and the constraints are known. If not all the variables are recognized, the "optimum" solution may be very bad. For instance, in laying out a rural highway, with much cut and fill in low-value land, there is good reason to optimize the cut and fill so that

volume cut = volume filled, and

volume cut = minimum, and

$\int V dx = \text{minimum}$, where x is the distance cut
volume V is moved to fill.

This may give a bad solution to the problem of laying out a road on a hillside for a new town. There, cost of cut and fill may be small compared to the value of the plots, and so then the optimization has to take into account the dimensions of plots adjacent to the road and their value.

7.3. Analysis Formulae and Inference Rules as Domain Switches

The data frame is a conceptual store or receptacle for the names and instantiated values of variables. The frame structure enables one to identify the variables in a frame, but not the relations between them or between variables in different frames. These relations are available

either as inference rules or as analysis formulae or analysis codes in computer programs.

The variables in any data frames may be from many domains. Some variables are expressed numerically, such as dimensions, masses, and electrical resistances; some are expressed by words, such as color, reliability, and safety rating.

The inference rules or numerical analyses may act within one domain (such as dimension or electrical properties) or may link together two or more domains (as in the example in Section 7.2.2).

Inference rules or numerical analyses may thus be thought of as entities that link together variables from one or more domains in one or more data frames. This is shown schematically in Figure 7d.

7.4. Constraint Decoupling Frames

7.4.1. Temporary Decoupling Frames

As mentioned in Section 3.3, it is often necessary to provide "working assumptions" for the design of different subsystems, in order to get a job done. This may arise if the production or construction procedure requires ordering or fabrication of a component before all the subsystems affected by that component are designed; or it may arise if a design manager wishes to avoid that one person or team wait for output from another person or team; or, if the problem is large (as it often is) and searching for a solution may take much time, the decoupling frame is used to decrease the amount of searching for a solution.

The "working assumptions" must be such that as the design progresses and more and more variables are instantiated, there will not be conflicts with the values of parameters already chosen. The "working assumptions" may be considered as extra constraint relations which are later subsumed by the design work and therefore are temporary. They must, however, be carefully chosen, or else as the design work evolves it will be found that decisions based on the working assumptions are wrong, and parts of the system must be redesigned.

The working assumptions are constraint decouplers, because their use enables work to continue without the evaluation of all the constraints otherwise necessary to progress in the design. They can be considered to emanate from special frames. A constraint-decoupling frame contains no data for instantiation, only terminals to data frames that are to be decoupled. The decoupling relation is the content of the decoupling frame. Figure 8 shows a decoupling frame used in a simple case to decouple foundation design from superstructure design.

7.4.2. Goal Specification as Node Decoupling

It was pointed out in Section 5 that the product of one process is the input or production machine for another process. For any product, the product specification is a set of constraints to that product designer; however, to the user of the product, the product specification is the instantiated frame of data for that product. The product specification is therefore a set of constraints that decouples the design work of the product designer from the product user.

Looking at the design work of the entire industrial economy, it can be seen that one designer's goal frame is another designer's ground (or bound or instantiated) frame. In this highly interconnected network, if decoupling were not achieved between nodes, design work would take a long time (as one designer waited for another's output) or else conflicting instantiations would occur within the total system. Specification of goal constraints is therefore a very important means of decoupling the work of different designers. These constraints can be represented in the data network by decoupling frames, which exist for a period of time as long as the product is in design, production, or use. These constraint decoupling frames have terminals that connect to frames of data, and their content is the values of the specified variables.

8 FEASIBILITY STUDY: ESTABLISHING A DESIGN CONCEPT

8.1. New Design and Redesign

Much design activity is related to improving the properties of existing products. This is called redesign, and is characterized here by a frame structure for the new product that is used almost unchanged from the old product.

The redesigned product will have some new or different constraints on the goal frame. As the goal frame is expanded into its subframes, most of the frames will be structurally unchanged, and most of the instantiations will probably also be unchanged. Some of the frames will be new, representing components not previously used. Redesign of an old product is therefore characterized by use of an existing frame network; the new frames, if they are generated, are fitted into this frame structure.

A new design requires that a frame structure be set up before instantiation of variables can progress. This is a very creative part of the design process and requires abilities as yet only vaguely understood.

8.2. The Process of Designing Data Frames

The creative part of a design is seen here as the process of choosing the right frame structure for the design. In order to do this, it may be that some constraints on the goal node in the frame net (which define the need to be satisfied -- see Figure 2) have to be deleted or changed and new constraints may have to be added. As pointed out in Section 3, when generating a new design it is in principle not possible to define all the design specifications ahead of time because some of the specifications themselves depend on the engineering concept to be developed.

The postulated model of engineering design can be used as a framework for understanding concept formation; it also points up the difficulties of doing such concept formation in a computer.

The starting point for the concept formation is an expressed need, which is a set of constraints (see Figure 9). From all the items of data available to him, the designer assembles a frame of data. This frame represents the most general level of the solution. The constraints are applied to the frame. If they cannot be satisfied, the designer tries to change the constraints. If that cannot be done, he assembles another data frame and tries again. When a frame is found that satisfies the constraints, the frame terminals are examined to find constraints not explicitly stated. Some of the terminals of the frame may thus be connected to irrelevant constraints, but all the terminals should be examined. If the new constraints lead to conflicting instantiations in the frame, either constraints or frame structure, or both, are changed.

If the goal frame survives this test, the next level of frame is examined, and so on. Implicit in the frame structure are the relations between each level and its subsystems, enabling the frames to be dealt with hierarchically.

The control strategy and the structure of the frames are inseparable. This model expresses the common experience that the inexperienced designer may not recognize some constraint relations, or variables in a frame, so that only after he is some way into the work will he find out about them. If the variables previously overlooked, or those referred to by previously overlooked constraints, may be dealt with separately from those already processed (because they form a partitionable or decomposable set of variables), then they can be evaluated. But if they interact with previously instantiated variables, then part (or much) of the design may have to be redone.

An experienced, or good designer, on the other hand, has the following characteristics:

- * He includes in the design solution novel components or configurations.

- * He foresees unusual constraints, thus saving valuable time.
- * He defines the problem so that a minimum of work is expended on nonapplicable solutions.

In other words, a good designer

- * Designs a goal frame with novel variables, not previously used for solving the problem he deals with.
- * Includes among the constraints applied to the goal frames some that operate on lower-level frames.
- * Defines the frames so that the network is hierarchic, with as few interactions across the tree as possible.

Achievement of such ability by a computer leads to a difficulty. To create a novel frame that suits the goal constraints, the computer must be able to combine different, independent, data entities, in order to apply the constraints. This is easily done for a small data base, but is done more quickly by a person. For a large data base, such frame building and testing against goal constraints takes much time. Furthermore, the computer's data base is bounded whereas the designer's responsibility is absolute, so that in principle he has to see to it that he accesses all relevant data.

This model therefore leads to the conclusion that there is no alternative to man-made (possibly computer-assisted) frame construction.

What principles could govern the design of a frame structure? If the names of all variables to be instantiated are known, how best to group them together?

There are two basic ways to arrange the frames, each of which is hierarchic. One is by domain, the other by subassembly. This will be demonstrated by considering the two alternatives for design of a calculator. Figure 6b showed a hierarchic arrangement of the data for the electronic domain. There are data from other domains, such as physical size and space allocation, heat generation and cooling, waterproofing, resistance to force and vibration, human engineering, safety, reliability, cost, time of delivery of components, and so on. These variables could be part of the frames laid out according to the hierarchy in the electronic domain. In considering whether to use this

approach, one must ask how many constraint relations would be needed across the tree, and up to how far across the tree such constraints would go (the range of the constraint). These same questions should be posed when considering organization of the data frames by physical subassembly.

In the days when transistors were first developed, the electronic designer could design his circuit to completion, then pass it on to the packaging, or mechanical, designer. With the high concentration of today's chips, and the high performance required of electronic instruments in withstanding environmental influences such as temperature, impact, dust, and humidity, such a decomposed approach, where electronic considerations are decoupled from mechanical considerations, is not possible. Constraints due to heat dissipation, waterproofing and impact or vibration must be taken into account from the very beginning. The conclusion, therefore, is that arrangement of frames by physical subsystems is more efficient than by electrical domain. The data defining the electronic circuit, if needed, can be obtained from the subsystem frames by extracting the data for the electronic domain only, together with the constraint relations for that domain.

Although generalizations cannot be made from one example, analysis of design problems in many fields suggests that it is often better to arrange data by physical subsystem, where each subsystem includes many domains, than by any one domain.

One may therefore establish two principles for frame generation.

(1) Each frame represents a physical subassembly, which is composed of subassemblies and is itself part of a higher-level subassembly. The goal frame is a subassembly for the system into which it eventually goes, but that relationship is neglected since the goal constraints are considered to stem from a decoupling frame.

(2) In the event that (1) above leads to the conclusion that some variable may be placed in one of several frames, the constraints operating on that variable should be examined, and the variable placed in the frame such that interaction by constraint relations across the tree is a minimal.

9 DESIGN DURING THE GELLED PHASE

In the gelled phase, the frame structure is organized and the variables to be instantiated in each frame are identified. In general, the gelled phase will exist for redesign of an old product, and so the new frames or variables required should be known after the design concept is established and the frames designed. The first time a product is designed, or if redesigned with insufficiently detailed planning, new variables and frames may appear as the work progresses. This can cause interactions across the frame tree and should be avoided.

As mentioned in Section 8, the frame should be constructed so that any external constraints to lower-level frames are known to the goal node at the time the goal node is instantiated. This permits the following control structure (see Figure 10).

(1) The constraints relevant to goal-node variables are applied, and the possible bindings required or excluded by those constraints are so marked.

(2) Some constraints at the goal node will be applicable to variables at more detailed levels than the goal. For these constraints, the names of possible bindings to the goal frame are examined to find which goal-node variable has as a possible lower-level binding the variable required by the constraint relation. The constraint is applied at its level, and the possible bindings along the path from it up to the goal node are evaluated and marked either as excluded or required.

(3) If one constraint requires a variable excluded by another constraint, a message is sent to the designer, who has to relax one of those constraints or create another frame (i.e., define a new binding that will allow the previously conflicting constraints to exist).

(4) The process in (2) may be short-circuited by a decoupling frame created by the designer; this frame will apply a goal-level constraint that includes the lower-level constraint, allowing work to progress more rapidly.

(5) After the goal node is dealt with, the next level of frame is dealt with.

(6) When dealing with any frame, the constraints are applied in the following order:

- (a) Node constraints, or absolute constraints -- for instance, <Steel thickness must exceed 0.5 inch> or <the pump used must be type T>
- (b) Arc constraints, between two frames -- for instance, "width of beam flange must equal width of column flange."

Mackworth (1977) has published algorithms that will achieve consistency of constraint relations (or discover inconsistencies) with a minimum of effort. When applying arc constraints, one must propagate them through the system until they induce no change. Rather than propagate each time through the whole network, thus checking much data unaffected by the constraint, one deals only with the relevant data.

- (c) Path constraints, for which a constraint on one arc is explicitly linked to another on another arc, and so on. This can be considered a general case of arc constraints.

10 ENGINEERING DESIGN IN THE FRAMEWORK OF THE MODEL

A number of examples and concepts in engineering design can be described in the language of the model presented here.

Consider one definition of an engineer, as set forth at a conference on training of engineers (Institution of Civil Engineers, London, 1971). *

The true professional exercises judgement in situations where there is no detailed information on the problem to be solved. He applies experience and technical training to decide in matters of production or design. This is different from a person who uses a book of methods or standards and hence cannot err.

The latter is usually called a technician. In other words, a technician can be responsible for instantiating variables in a given system of frames and constraint relations, but the responsibility for design of the frames, for inclusion of all relevant variables and constraint relations, falls on the engineer.

For that reason, a computer can never replace a designer, as sometimes suggested, since a computer's data base is bounded, whereas a designer has the responsibility that he does indeed use all relevant information.

An interesting question arises, whether a product specification may be used as the set of constraint relations for the goal node in the frame net. From the limited research that has been done on the logical structure of specifications (Fenves and Norabhoompipat 1978) it appears that specifications usually show the following characteristics:

- * Use of words or expressions without precisely specified meanings, so that judgment and discussion are used when translating the specifications into practice.

* This definition has been translated twice before quotation here. The original statement was probably differently worded.

- * Provisions for different ranges of various variables, which overlap at the ends of the ranges and conflict.
- * Various situations or regions of variables not covered by specifications.

Product specifications as they are written today can form part of the goal specification; even then they need to be checked for logical inconsistencies.

The process whereby design concepts are established is of course very important, and is rooted not only in technical considerations, but also in concepts of social organization and responsibility. For instance, considerations of pollution and environment are today much more prominent than several decades ago, not only because there is more pollution, but also because there is pressure from society. Head (1978) describes the design processes for weapons in the United States and the USSR, particularly the processes leading to of a design concept. He states that in the USSR, design concepts are established by a hierarchical system of defined responsibilities, and there are strong tendencies toward product evolution (i.e., improvement by redesign) and multiple use of components; in the United States, the process of establishing design concepts is far more diffuse, less structured, and integrates opinions and proposals from a wide range of groups, with the tendency toward developing whole new systems.

Most engineering design is redesign; that is, the structure of data frames and constraint relations is used, with changes made to some of the frames in this structure. New data frames are added only if the changes to the old data structure do not conflict with whatever constraints are applied (such as the need to use this or that component for economic or administrative reasons). Hopper (1978) shows how the requirement for redesign can constrain a solution for a new part of an old design. Referring to aircraft design, he states:

The current interest in head up display (HUD)...has led to consideration of overhead installation since the prime real estate on the instrument panel is already allocated and difficult to disturb without potential certification problems.

This situation will not change so long as HUD's are added as afterthoughts to the cockpit instrumentation. Perhaps one of these days an aircraft manufacturer will design a new airplane with the HUD located where it belongs right from the start.

11 SOME EXISTING WORK IN COMPUTER-AIDED DESIGN
REVIEWED IN THE LIGHT OF THE MODEL

A number of programs have been written and reports have been made on the problem of space allocation in printed-circuit boards (PCBs) for wires and conduction paths, and in architecture for locating rooms and furniture. It is interesting to note (as reported, for instance, at the CAD78 conference in Brighton, England) that programs for PCB design are part of accepted design practice, whereas programs for architectural layout are regarded as experimental and often give results no better than those of people (Cross 1978). When such a program is used in practice, as in Hashimshony et al. (1978), the output is analyzed and changed by the designer before being used.

This difference in utility is easily analyzed in terms of the present model. For a printed circuit board, the spatial domain is decoupled from other domains in the frame; a layout solution for the decoupled spatial domain is therefore acceptable. In architecture, the space-planning program makes use of spatial relations between items (a desk may not block a door) and functional relations (deriving, for instance, from pedestrian traffic flow). These functional relations may not be completely identified. Also, the desired result is affected by considerations of human engineering such as views from windows, and so on. The spatial layout program deals with the spatial domain decoupled from the aesthetic domain; in order to succeed in architecture, however, it should not be decoupled. Unfortunately, for the "instinctive" domain of the architect, relevant variables and the relations between them and the spatial domain are not yet sufficiently understood. If the success of automatic printed-circuit-board layout depended on the aesthetics of the board, it is doubtful whether space allocation for PCBs would have succeeded as well as it has.

It is interesting to note that a small number of contextual relations can add greatly to the utility of a program. Liardet et al (1978) describe an architectural site layout program, with the following capability:

- * The program recognizes the image of a house, relative to the image of a site boundary or path.
- * The image of a house can only be rotated or translated, but not distorted.

The layout is adjusted subject to the conditions:

- * All houses must be included within site boundaries.
- * Paths may overlap gardens.
- * Houses must not overlap each other.

This program may be thought of as having data frames with labels for house, path and site, and constraint relations for those labels. Although the constraint relations are few, the results obtained are much more useful than if the designer had to evaluate the constraints and edit the layout himself (Lairdet et al. 1978).

Various three-dimensional geometric layout systems for aircraft or chemical plants inform the designer of space conflicts. The universal spatial constraint which provides this very useful information, can be expressed as

<for any item a, for all items b, ($\sim(x(a)=x(b))$).or.

$\sim(y(a)=y(b))$).or. $\sim(z(a)=z(b))$).or. $\sim(t(a)=t(b))$ >

here x, y, and z are space coordinates, t is time, and $\sim(a = b)$.

12 RESEARCH IN ARTIFICIAL INTELLIGENCE
IN THE LIGHT OF THE MODEL

There are in the technical literature various statements about design, and descriptions of computer programs that do design problem-solving.

"Design is redesign" is sometimes stated. It is true that the largest volume of design work is redesign, which means instantiating variables in an existing net of data frames and constraint relations. However, the activity that may be defined as new design, which starts with design of the data frames, although of smaller quantity than redesign is important, because that is where the frontiers of technological capability are pushed forward.

"Top down or bottom up?" is sometimes posed as a question. "Top down" is taken to mean that one starts with a general concept, but no definition of detail, then fills out the detail later. "Bottom up" means that one starts with detailed components, without knowing how they will form a system. (Heath Robinson cartoons are bottom-up designs in this sense.) Design is neither of the two extremes. Design starts off with a system specification, which often will include the requirement that certain components be used. The model proposed here makes use of that by recognizing the constraints applied at the detailed levels as part of the goal frame. Design can then proceed from the general to the detailed, with the known detail being part of the constraints applied to the goal frame.

"Design is debugging" is sometimes stated. By "debugging" is meant (a) discovery of a conflict between two or more constraint relations, and (b) resolution of the conflict by changing one or more constraint relations, and/or by changing one or more data frames, and/or by creating one or more data frames. This is certainly part of the design

process. Discovering a constraint conflict and resolving it is a common way of solving design problems (Akin 1978a, Henrion 1978).

A number of programs written for simulating the process of design correspond to particular implementations of the scheme outlined here.

Freeman and Newell (1971) developed a program for functional reasoning in design. The data for this program were the purpose of each of a number of components, and a list of functions or purposes provided by the components. A goal could be satisfied by finding that set of components that satisfied the goal conditions and the functional requirement conditions between them. The functions served by and required by any component can be thought of as terminals or labels on the frame for that component. The frames in this context are ground level frames, since the components exist and cannot be changed by the design. The design process then starts with the goal node constraints and looks for a set of frames that will satisfy both the goal node constraints and the constraints between the frames.

The TROPIC system of Latombe (1976) represents an extension of the functional approach of Freeman and Newell (1971) in the direction of the scheme suggested here. Latombe defines hierarchic levels of abstraction for representation of components. The hierarchic organization is represented by hierarchic relations; other relations are represented as associative relations. The hierarchic and associative relations are embedded in the data structure -- that is, linked to the frames of data. A list of possible values of properties of each object is in the embedded data. The designer specifies the problem by inputting a set of relations relevant to any of the objects in the problem domain; he also can input "advice" on a list of relations.

For instance, in designing a tower, advice may be, "If the tower is to be more than 150 feet high, it is likely that the cross section of the blocks of its pillar will have to be round." The program searches for a set of objects and their properties that will satisfy the problem definition and the embedded relations between objects. At any point, if the current state of the search is such that a piece of advice is

relevant, the advice is used. The advice, therefore, plays the part of a constraint decoupling (or short circuiting), as described in section 7.3.1, which saves searching when looking for a solution.

Stallman and Sussman (1977) have written a program that designs electronic circuits by using equations and inference relations as constraint relations, which are propagated. The program keeps track of the paths followed in the inference net so that it can give the reason for any decision or for requesting any data. The program works entirely in the electronic circuit analysis domain, but does not include data from other domains such as heat transfer or packaging. To deal in reasonable time with nontrivial problems, Stallman and Sussman grouped several circuit components into a single entity (a frame, in the concept of this paper) and applied constraints to this more general frame, which included the constraint relations for the individual components. This is a use both of a hierarchic frame data structure and temporary decoupling constraints. They also evaluated constraints in an order specified by the programmer, so that those constraints more likely to find contradictions were dealt with earlier.

Another observation made by Stallman and Sussman is the importance of dividing the circuit into modules, with rigidly defined interfaces, so that the effects of any one change will propagate only locally. This is a use of node decoupling frames in the data net. Each decoupled part of the net could in principle be designed by a different designer. For the relatively small (but real) problems they dealt with, this decoupling was important, and so it would be important in design in general, as mentioned in Section 7.3.

Rieger (1976), in explaining an approach to understanding knowledge in language and problem solving, mentions the importance both of hierarchic ordering of knowledge and of decoupling constraints when dealing with that knowledge.

13 CONCLUSIONS

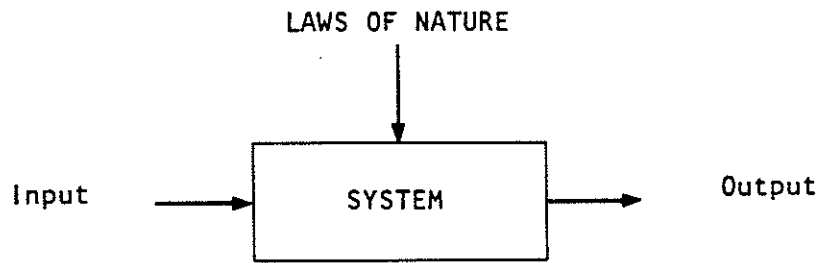
Engineering design can be viewed as a process of establishing data frames, then instantiating data in the frames by constraint propagation. The data include both numeric and nonnumeric information, and constraint relations are both numerical analysis formulae (or computer codes) and inference rules.

Most engineering design is within an existing network of frames; such activity is called redesign. Here, new instantiations or new data frames do not lead to interactions such that the old frame structure is rendered invalid.

Frames of data are organized in hierarchies of generality. The relations between variables and their lower-level possible instantiations are known. Goal-node constraints include constraints applied to variables at more detailed levels; passing these constraints through the goal node permits top down control of constraint propagation.

Temporary decoupling constraints, which short circuit many more detailed constraints, are important parts of the network. However, if the decoupling constraints do not include all the relations they temporarily decouple, contradictions may develop.

Constraints on the goal node, which are the requirement definitions for the design, are decoupling constraints between the designer and user of a product. One designer's goal node is therefore another designer's ground node, and designed products are themselves networked together, with the product specifications serving as node decouplers between them.



Given	Required	Activity
INPUT, OUTPUT, SYSTEM	LAWS OF NATURE	SCIENCE
INPUT, SYSTEM, LAWS OF NATURE	OUTPUT	ANALYSIS
INPUT, OUTPUT, LAWS OF NATURE	SYSTEM	DESIGN

FIGURE 1 SCIENCE, ANALYSIS, AND DESIGN

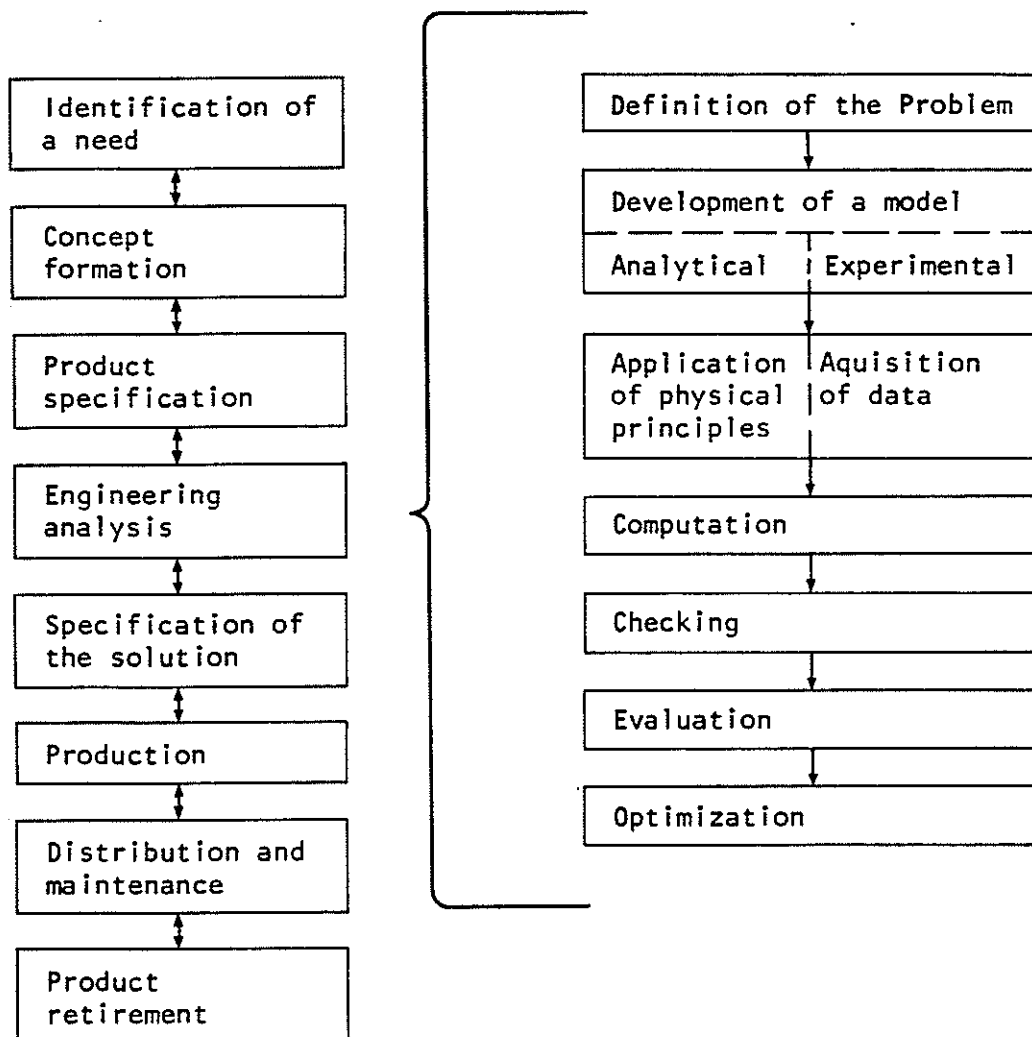


FIGURE 2 THE DESIGN PROCESS

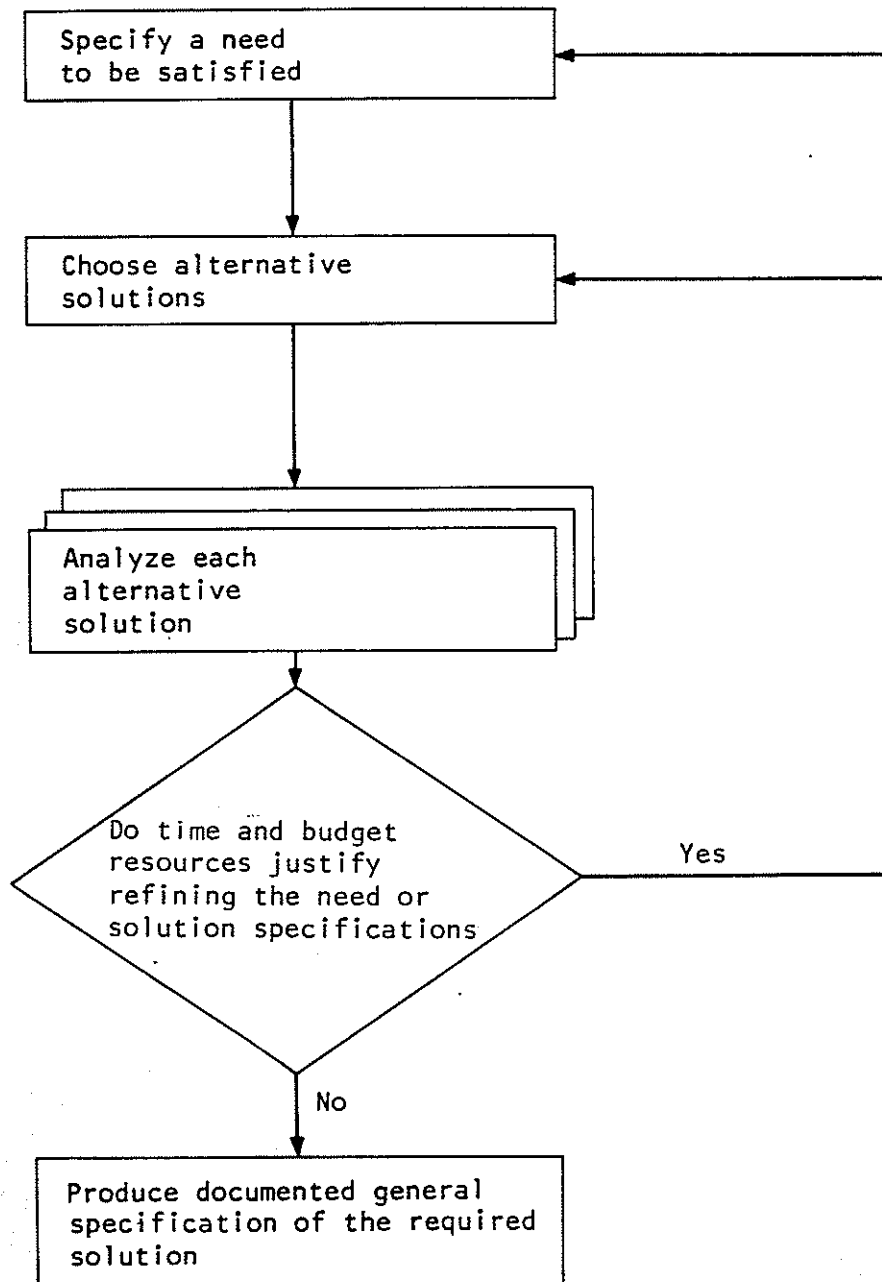


FIGURE 3 FEASIBILITY STUDY



FIGURE 4 A SYSTEM THAT PRODUCES A PHYSICAL PRODUCT

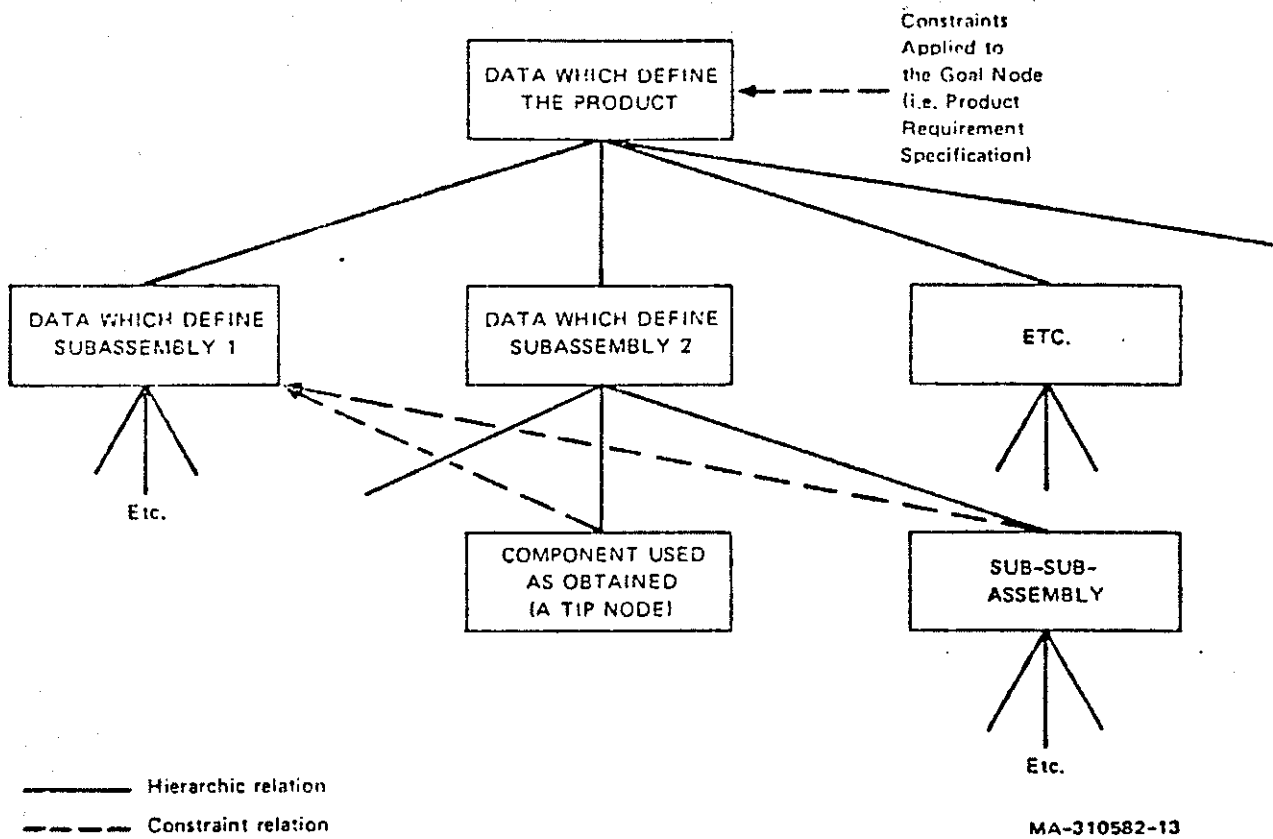


FIGURE 5 HIERARCHIC ARRANGEMENT OF DATA FRAMES FOR A PRODUCT

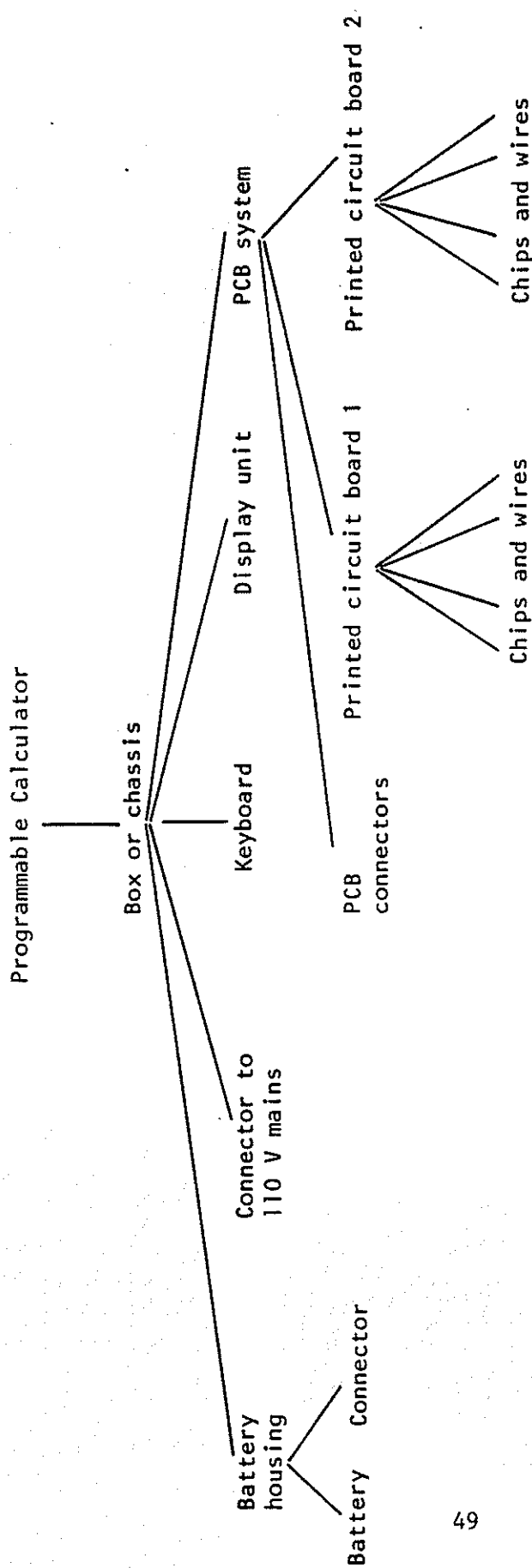


FIGURE 6a COMPONENTS OF A PROGRAMMABLE CALCULATOR SEEN FROM THE PHYSICAL-LAYOUT VIEWPOINT
 The data are hierarchic. Each tip node is for a component used as obtained. Relations in the electronic domain (e.g., battery to display unit) cut across the tree structure.

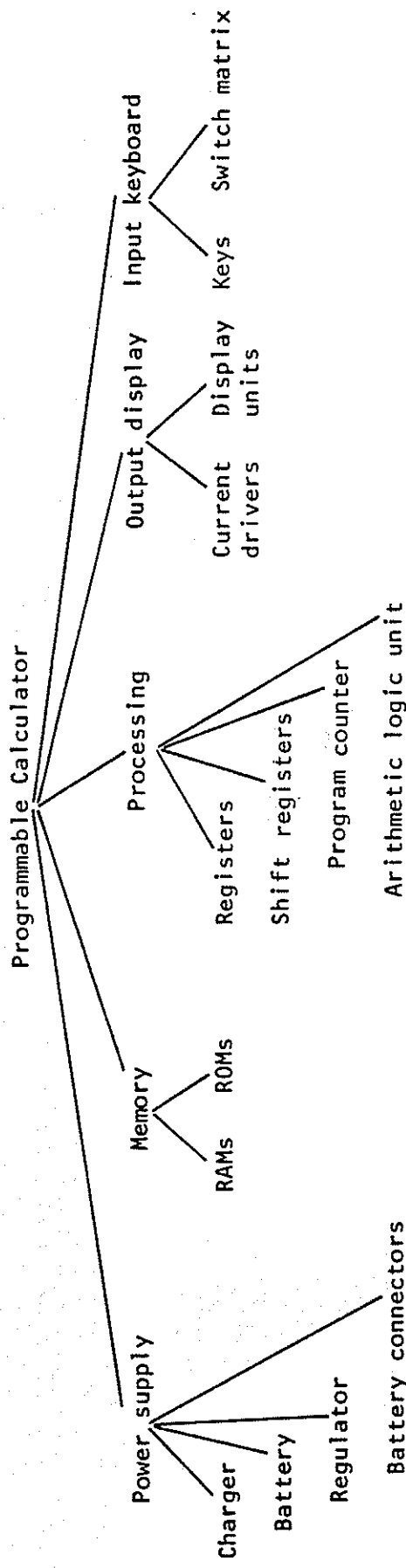


FIGURE 6b COMPONENTS OF A PROGRAMMABLE CALCULATOR SEEN FROM THE DOMAIN-SPECIFIC VIEWPOINT

The domain shown is electronic. The data are hierarchic. Spatial or other nonelectronic relations cut across the tree structure of the data.

LABEL	POSSIBLE VALUES
Environment	Desert, humid, seaside, indoor, . . .
Loads	Specification A, B, C
Type	Arch, suspension, truss, girder, . . .
Material	Steel, aluminum, wood, concrete, . . .
Foundations	Driven-piles, piers, caisson,
.	
.	
.	
.	
.	

FIGURE 7a PART OF A DATA FRAME FOR A BRIDGE

LABEL	SYMBOL	POSSIBLE ORDERED VALUES		
Material-name		Structural-steel-to-ASTM-A7		
Section-name		36x16-1/2x300	27x14x177	14x10x68
Depth	h inch	36.72	27.31	14.06
Width	b inch	16.655	14.090	10.04
Flange thickness	t inch	1.68	1.19	0.718
Web thickness	w inch	0.945	0.725	0.418
Weight	W lb/ft	300	177	68
Area	A in ²	88.17	52.10	20.0
Max. mom inertia	I _x in ⁴	20290	6279	724
Min. mom inertia	I _y in ⁴	1225	519	121
Yield stress	f _y lb/in ²	33000	33000	33000

FIGURE 7b PART OF A DATA FRAME FOR A BEAM

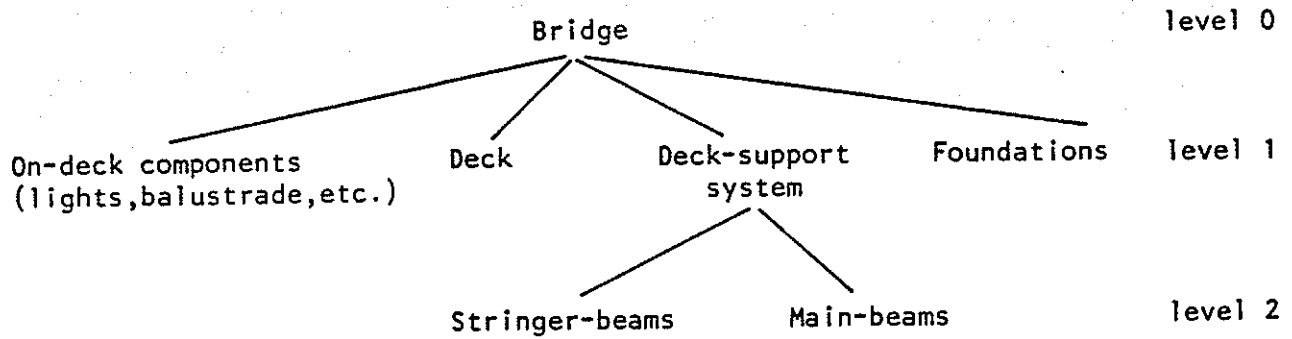


FIGURE 7c A POSSIBLE HIERARCHIC EXPANSION OF A DATA FRAME FOR A BRIDGE, SHOWING HOW A DATA FRAME FOR A BEAM FORMS PART OF THE BRIDGE

Different ways of arranging hierarchic relations are possible. For different fields (bridges, office blocks, etc.), or for different points of view (construction, strength analysis, etc.), different hierarchic relations will be more suitable.

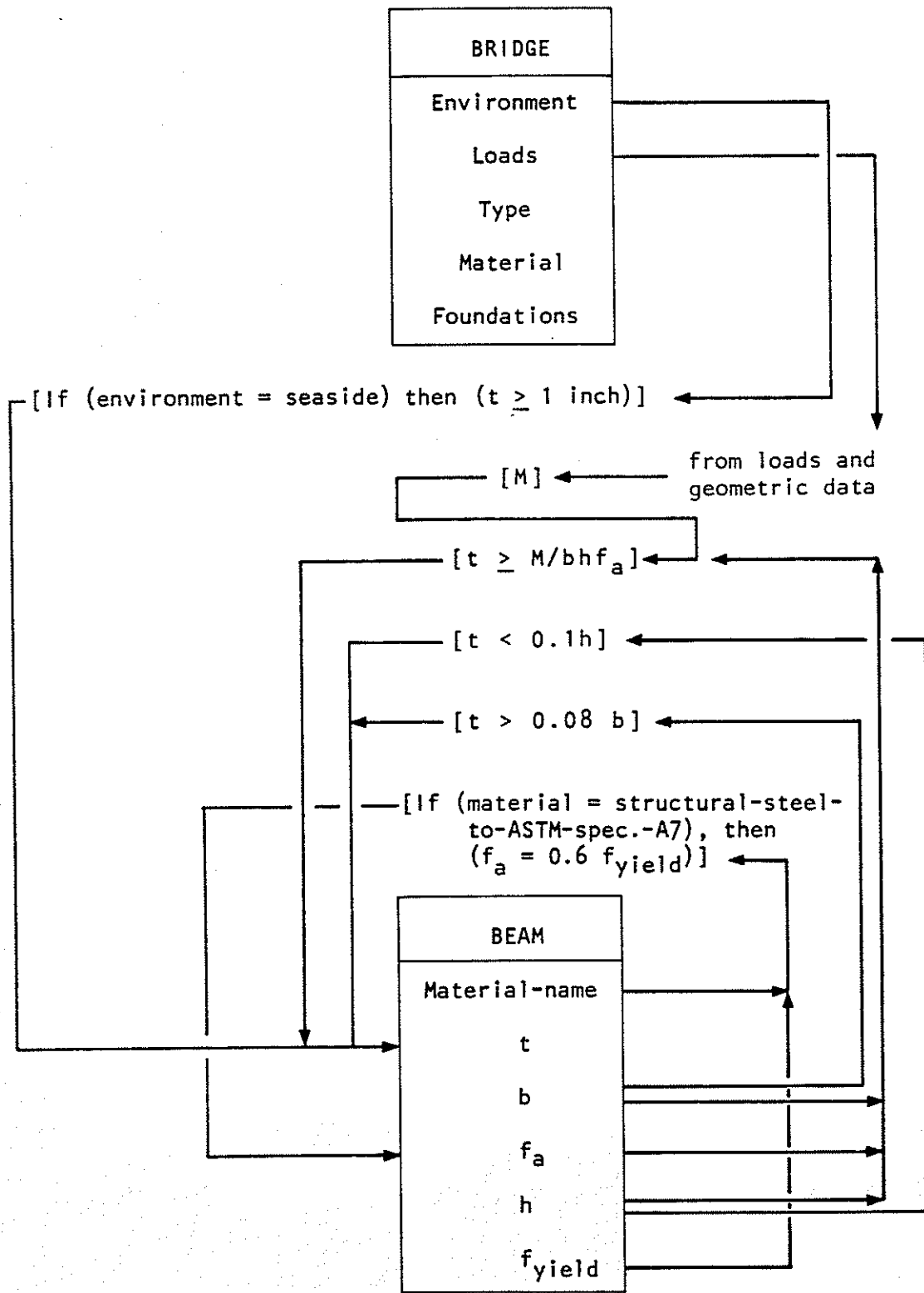
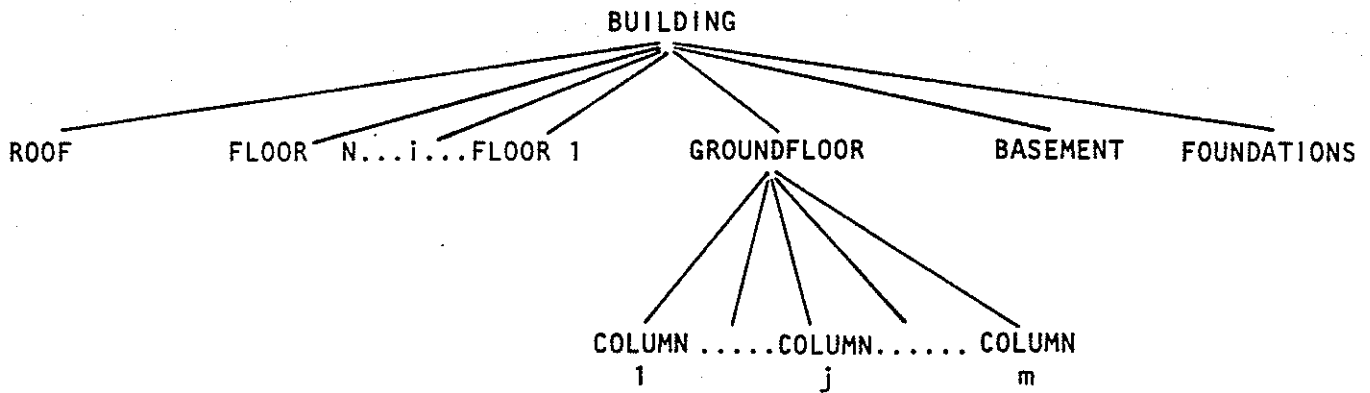


FIGURE 7d PARTIAL INFERENCE NET USED TO INSTANTIATE DATA IN THE DATA FRAME FOR THE BEAM



PARTIAL FRAME FOR FOUNDATION j

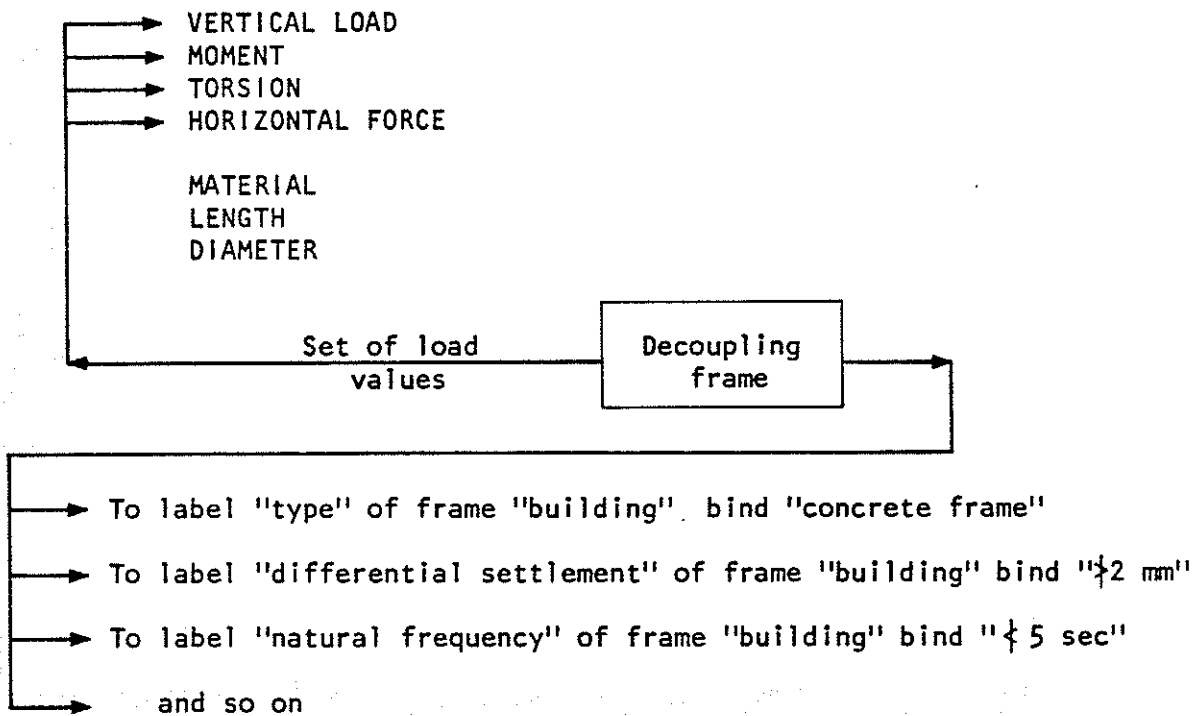
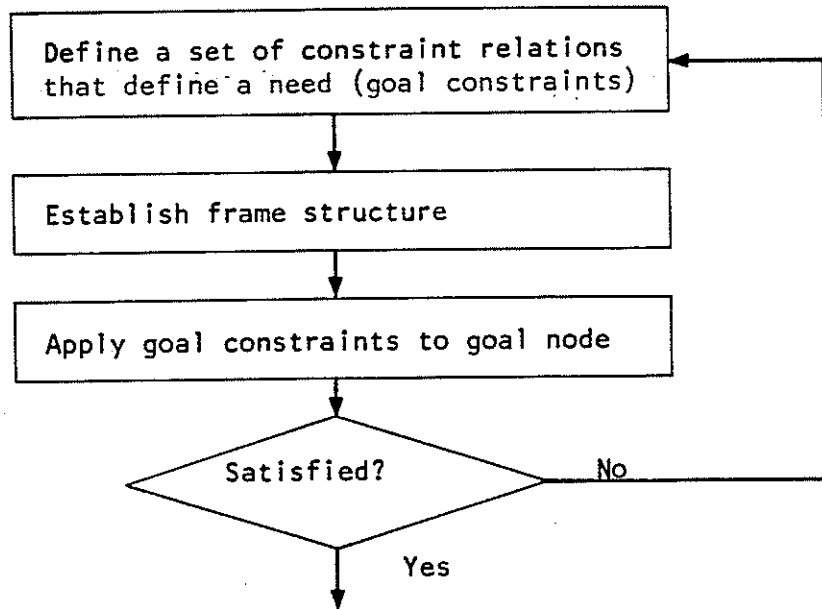


FIGURE 8 TEMPORARY DECOUPLING



Frame structure is established.
Proceed as for gelled phase (Figure 10)

FIGURE 9 ESTABLISHMENT OF A FRAME STRUCTURE

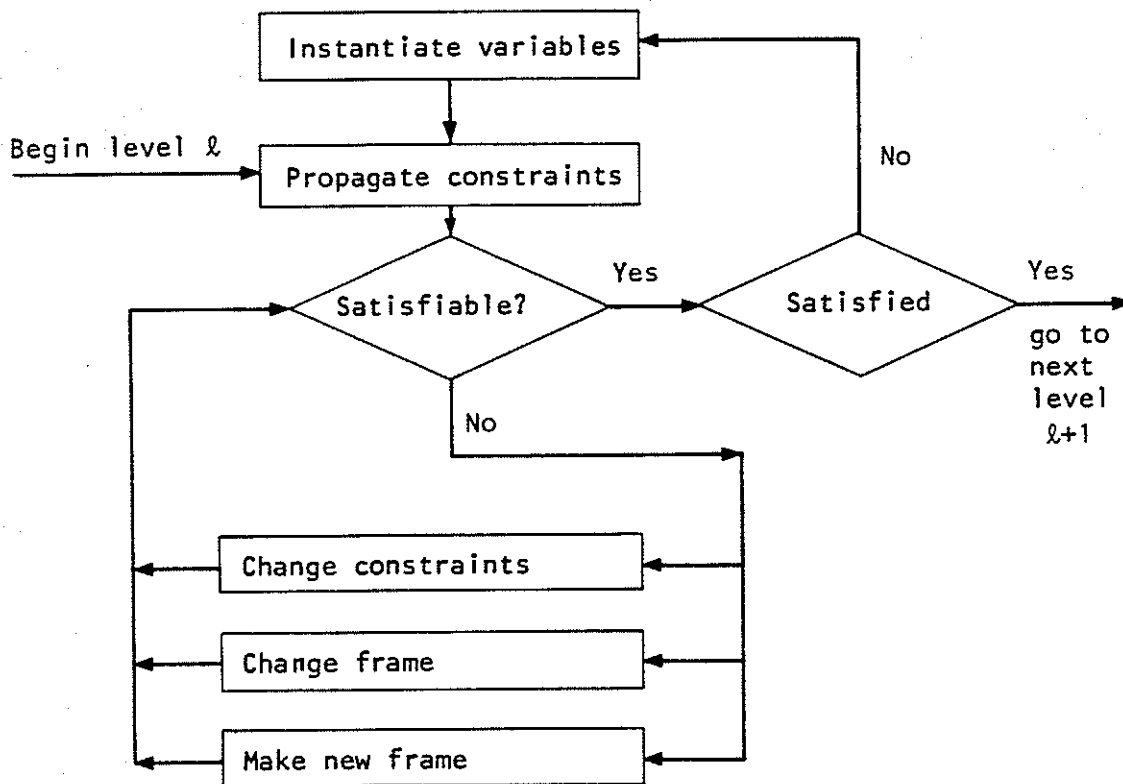


FIGURE 10 INSTANTIATION OF DATA FRAMES
IN THE GELLED PHASE

REFERENCES

- Akin, O. "How Do Architects Design?" Proc. IFIP 5.2, Working Conference on Artificial Intelligence and Pattern Recognition in Computer-aided Design, Grenoble, North Holland Press, 1978a.
- Akin, O. "AIM: Architectural Inference Maker," CAD 78 Conference, Brighton, IPC Press, Guildford, England, 1978b.
- American Institute of Steel Construction, "Manual of Steel Structures", 1963.
- Cross, N. "Assessing Computer-aided Architectural Design Systems," CAD78 Conference, Brighton, IPC Press, Guildford, England, 1978.
- Dixon, J. R. Design Engineering: Inventiveness, Analysis and Decision-making, McGraw-Hill, 1966.
- Fenves, S. J. and Norabhoompipat, T. "Potentials for Artificial Intelligence Applications in Structural Engineering Design and Detailing," Proc. IFIP WG 5.2, Working Conference on Artificial Intelligence and Pattern Recognition in Computer-Aided Design, Grenoble, North Holland, 1978.
- Freeman, P. and Newell, A. "A Model for Functional Reasoning in Design," International Joint Conference on Artificial Intelligence, 1971.
- Hashimshony, E., Shaviv, E. and Wachman, A. "A Decomposition Model as a Tool for Evaluation of Multi-cell Solutions: A Case Study," CAD78 Conference, Brighton, IPC Press, Guildford, England, 1978.
- Head, R. G. "Technology and the Military Balance," Foreign Affairs, Vol. 56, No. 3, pp. 544-563, April 1978.
- Henrion, M. "Automatic Space-Planning: A Postmortem?" Proc. IFIP WG 5.2, Working Conference on Artificial Intelligence and Pattern Recognition in Computer-Aided Design, Grenoble, North Holland, 1978.
- Hopper, L. W. Letter to the editor of Aviation Week and Space Technology, Vol. 108, No. 16, April 17, 1978.
- Kronman, J. "Wiring Harness Design and Fabrication for the L-1011," Computer Aided Design, Vol. 10, No. 2, pp. 141-145, March 1978.

- Latombe, J. C. "Artificial Intelligence in Computer Aided Design: the TROPIC System," Artificial Intelligence Center Technical Note 125, Stanford Research Institute, Menlo Park, California, 1976.
- Liardet, M., Holmes, C., and Rosenthal, D. "Input to CAD Systems: Two Practical Examples," Proc. IFIP WG 5.2. Working Conference on Artificial Intelligence and Pattern Recognition in Computer-Aided Design, Grenoble, North Holland, 1978.
- Mackworth, A. "Consistency in Constraint Propagation," Artificial Intelligence, Vol. 8, No. 1, pp. 99 - 118, 1977.
- Martin, J. Computer Data Base Organization, Prentice Hall, Englewood Cliffs, New Jersey, 1977.
- Newell, A. and Simon, H. A., Human Problem Solving, Prentice Hall, Englewood Cliffs, New Jersey, 1972.
- Ostrofsky, B., Design Planning and Development Methodology, Prentice Hall, Englewood Cliffs, New Jersey, 1977.
- Rieger, C., "An Organization of Knowledge for Problem-Solving and Language Comprehension," Artificial Intelligence, Vol. 7, pp. 89-127, 1976.
- Simon, H. A., The Sciences of the Artificial, MIT Press, Cambridge, Massachusetts, 1969.
- Stallman, R. M., and Sussman, G. J., "Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis," Artificial Intelligence, Vol. 9, pp. 135-196, 1977.
- Winston, P., Artificial Intelligence, Addison Wesley, 1977.