

Function sharing in mechanical design

Karl T. Ulrich and Warren P. Seering

Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Function sharing is the simultaneous implementation of several functions by a single structural element. This article describes how the idea of function sharing can be used in a computational design procedure that produces efficient designs from modular designs. These ideas have been implemented as a computer program for the domain of mechanical devices that can be described functionally as a network of lumped-parameter idealized elements.

Keywords: mechanical design, design theory, design automation

Function sharing in mechanical design is the simultaneous implementation in an artefact of several functions by a single structural element. Consider, for example, the difference between the devices shown in Figure 1. Although the two are functionally similar, the upper device is much more efficient because each structural element implements several functions. This article describes how function sharing can be achieved by a computational design procedure in order to produce efficient designs from modular designs.

The concept of function sharing

In synthesizing a design, engineers represent an artefact in terms of its constituent functional elements, its *schematic description*. Designers must also represent an artefact in terms of its constituent structural elements, its *physical description*. In the case of computational design systems, these representations will often be explicit and distinct, while in human design systems the representa-

tions are often informal and interleaved. For a given physical and schematic description of a device, there will be a correspondence between elements in each. Function sharing is a mapping from more than one element in a schematic description to a single element in a physical description. In the case of the devices shown in Figure 1, some of the functional elements in an imagined schematic language might be *cutter, actuator, finger interface, bearing and key ring interface*. The structural elements in a physical language might be the geometrical descriptions and material properties of each separate part.

Function sharing is important

Function sharing is important for three reasons. First, designs that exhibit function sharing are in most respects better designs than the non-function-sharing counterpart. Second, a design simplification procedure that results in function sharing allows the designer to think in a modular, decomposed fashion, with the option of subsequently processing a design to make it more efficient. Third, function sharing is often viewed as one of the sources of novelty or interest in mechanical design; so

A short version of this paper appeared in the proceedings of the *Seventh National Conference on Artificial Intelligence*, held in August 1988 at St Paul, MN, USA.

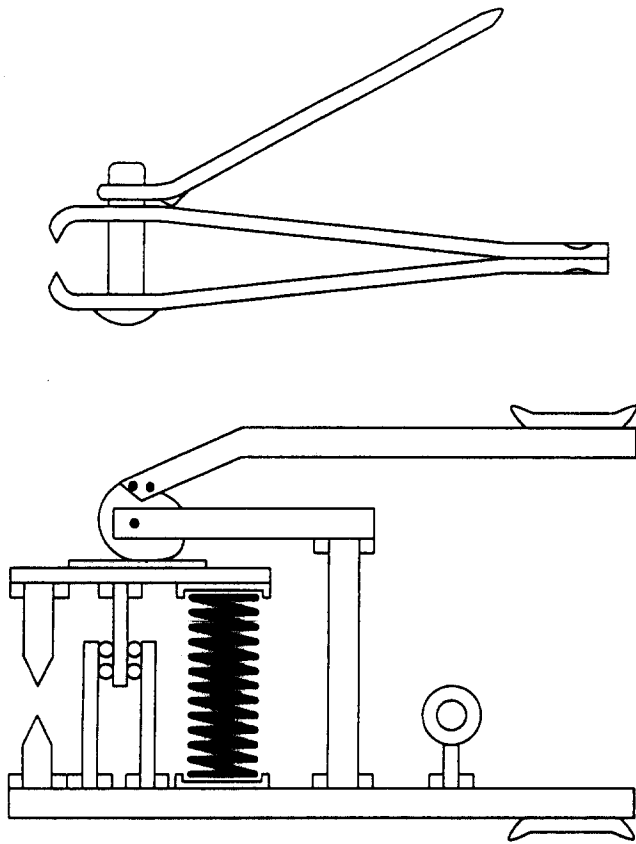


Figure 1. Example of function sharing

an understanding of function sharing is in part an understanding of novelty.

If automobiles were designed without function sharing they would be relatively large, heavy, expensive and unreliable. But because elements like the sheet-metal body perform many functions (electrical ground, structural support, aerodynamic faring, weather protection and aesthetics, among others) automobiles can be manufactured relatively inexpensively and can perform relatively well. In fact, most mass-produced devices exhibit function sharing. The leads on integrated circuit packages are electrical conductors, thermal conductors and attachment points for the circuit board. The physical structure of a styrofoam coffee cup holds liquid, insulates, provides a graspable surface and rests stably on a table. As a general rule, mass-produced designs that exhibit function sharing are better than those that do not exhibit function sharing. Function sharing contributes to design quality in two major ways. First, designs exhibiting function sharing will generally be less expensive to produce than designs that do not exhibit function sharing, as a result of fewer parts, easier assembly and less required adjustment and maintenance. Second, designs that share function generally perform better than those that do not, resulting from decreased size and weight. On the other hand, function sharing is generally a poor design strategy for research devices and prototypes where debugging, adjustment and diagnosis are very important, because function sharing often causes device performance parameters to be coupled in complicated ways to

device physical parameters. While function sharing may be an important design strategy for a ballpoint pen or an electric drill, it is a potentially disadvantageous strategy for a steel rolling mill or a laboratory instrument. For the purposes of this article, we assume that a computer system able to generate design descriptions that exhibit function sharing is a valuable tool.

Most designs evolve from being modular to being integrated. This is partly because it is easier to think about new problems if they are decomposed in a modular fashion, and partly because in the initial stages of product development engineers want to be able to work independently on different aspects of a design. The function sharing procedure described in this article is one way of transforming modular designs into integrated designs. In tandem with a procedure for generating the initial modular descriptions of designs¹, the procedure allows a design system to transform a behavioural specification into an efficient physical description.

We are interested in innovative design. Function sharing is part of the perception of novelty, simplicity or interest in a device. To the extent that function sharing can be understood, one of the components of novel mechanical design can be understood.

Why function sharing is possible

Of the infinite physical properties of a structural element, only a small set are relevant to the behaviour the designer intends for that element. In addition to the primary properties of a structural element that provide that element's intended function, there are many secondary properties that are incidental to those that implement the intended function. The key idea that allows function sharing to be achieved by a design procedure is that these secondary properties of structural elements can be exploited. By recognizing and exploiting secondary properties of one element, neighbouring elements can be eliminated from the design.

For example, a modular design of an automobile would include a ground wire running from the tail light to the battery. By recognizing that there is already an element (the steel automobile body) that travels from the tail light to the battery, and that this element has the secondary property that it conducts electricity, the ground wire can be eliminated. In this example, this simplification is possible because steel has many properties other than high stiffness, high strength, ductility and low cost (ostensibly the reasons for using steel for car bodies). Performing this reasoning requires a physical representation of the design, and an ability to recognize secondary properties of regions of the physical description.

Summary of function sharing procedure

The procedure for achieving function sharing consists of three steps:

- (1) a structural element is deleted from the physical description
- (2) alternative features in the physical description that can potentially implement the function of the deleted element are identified
- (3) the identified features are modified to accentuate their desirable secondary properties

This procedure is illustrated in Figure 2. The initial design is a lamp fixture hung by a piece of link chain, through which the electrical cord is woven. In order to simplify the lamp design, an element (the chain) is deleted from the design. Next, the tensile properties of the electrical cord are recognized. Finally, these properties are exploited by making the cord thicker. Through this three-step process the design is simplified.

DOMAIN DESCRIPTION AND REPRESENTATION

Dynamic systems

As a domain for exploring function sharing, we have chosen mechanical devices whose primary behaviour can be described by a differential equation relating an input and output quantity and whose schematic description can be expressed as a network of lumped-parameter idealized elements. The computer program that implements the

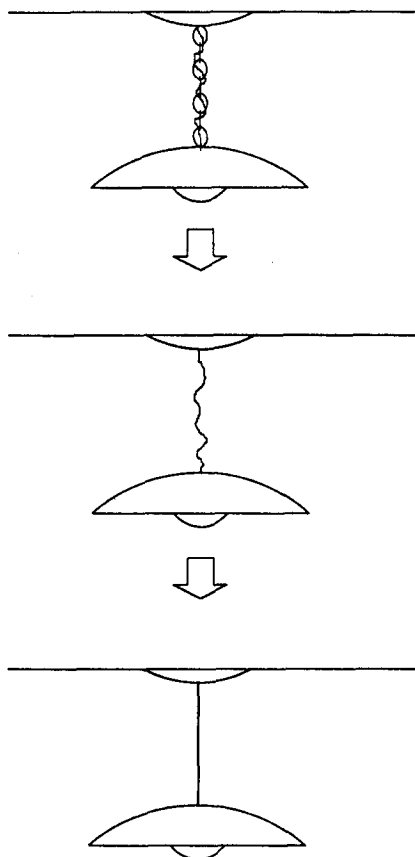


Figure 2. Illustration of function sharing procedure

function sharing procedure is further limited to devices that can be described with fluid-mechanical elements and mechanical-translational elements. Such devices include pressure gauges, accelerometers, force transducers and pneumatic cylinders. We call this domain *dynamic systems*. Examples of diverse designs that are *not* in this class are conveyer systems, ball bearings and computer displays (although these systems can be described with differential equations and networks, such descriptions do not characterize their primary behaviour). We have chosen the dynamic systems domain because there is a well-developed schematic language for the domain, and because many of the geometrical issues can be understood in two dimensions.

Physical representation

We have chosen a two-and-one-half dimensional geometry to describe devices physically. In this representation, physical descriptions of designs consist of a collection of structural elements (these can be thought of as design components like piston-cylinders or springs), which are in turn built from orthogonally connected rectangular-prismatic sections of material (these rectangular-prismatic sections are the primitive physical building blocks of the system). Figure 3 shows the physical description of a piston-cylinder structural element. Figure 4 shows the top view of the physical description of a rate-of-pressure indicator containing the piston-cylinder structural element.

The two-and-one-half dimensional representation is required rather than a two-dimensional representation to allow fluid to flow past obstacles in the design. For example, if the piston-cylinder in Figure 3 were only two-dimensional, the two cavities above and below the piston rod would not be in fluid communication and the piston would not function as expected. By adding an additional half dimension (thickness) to the design description, the upper and lower cavities can be connected as long as the piston rod is only half as thick as the cylinder.

In this two-and-one-half dimensional geometry, Newtonian physics applies as if the device were fully three-dimensional if one imagines the device to be sand-

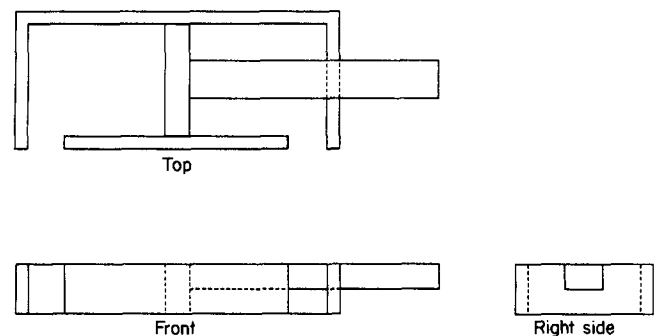


Figure 3. A piston-cylinder represented as a collection of orthogonally configured rectangular-prismatic sections

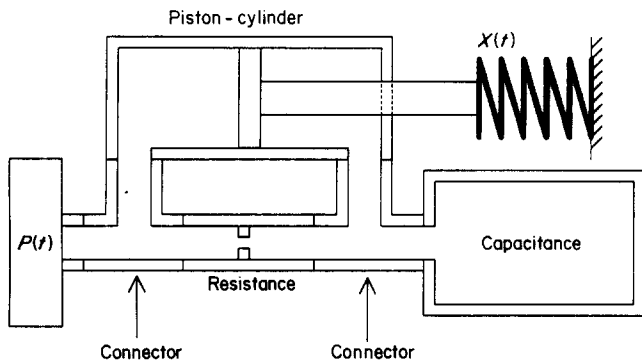


Figure 4. Example of a physical description of a design

wicked between two infinitely stiff and strong, frictionless plates. This particular physical representation was chosen to simplify the computational geometry problems while still maintaining the applicability of Newtonian physics.

In addition to specifying the x and y locations and dimensions of each rectangular-prismatic section of the design, a limited set of material properties are also specified. The material of each section is specified to be either *stiff* or *flexible*, and of *high* or *low* density. Each section is also specified as either attached to or free from the imaginary plates that sandwich the design. Designs consisting of fluid-mechanical elements also include a working fluid specification (either *gas* or *liquid*). All design descriptions contain a specification of an external environment for the design (also either *gas* or *liquid*).

Nominal functional specification

In the dynamic systems domain, the schematic description of a device can be expressed as a network of idealized lumped-parameter functional elements. The schematic description can be thought of as a generalized circuit diagram, so the behaviour of the entire device can be derived from the behaviour of the individual idealized elements and the network laws. In addition to specifying the strictly physical properties of a design, the function sharing procedure requires a specification of the intended function of each structural element. This specification is an explicit linking of functional elements from the schematic description to regions of the physical design description. A function specification includes the name of the function and one or two reference points in the design. For example, the specification of a fluid resistance would indicate first, that the function is *fluid resistance*; second, that the resistance is located between two points identified with x - y coordinates. The nominal functions of all structural elements in a physical description are specified similarly.

In addition to the lumped-parameter functions, two special functional elements are also necessary: *connection* and *ground*. These are used to specify how regions of the physical description connect the other functional elements together.

FUNCTION SHARING PROCEDURE

The input to the function sharing procedure is a physical description of a device augmented with a specification of the intended function of each structural element; the output is also a physical description. The objective of the procedure is to simplify the initial physical description in order to reduce the number of structural elements.

The function sharing procedure consists of three steps: (1) a structural element is deleted from the design; (2) geometrical features that can potentially implement the function of the deleted element are found; (3) modifications are made to the design to exploit the secondary properties of the features found in step 2. After illustrating these steps with an example, we explain each step for the dynamic systems domain.

Example of function sharing procedure

Figures 5 and 6 display an example of the function sharing procedure in the dynamic systems domain. The

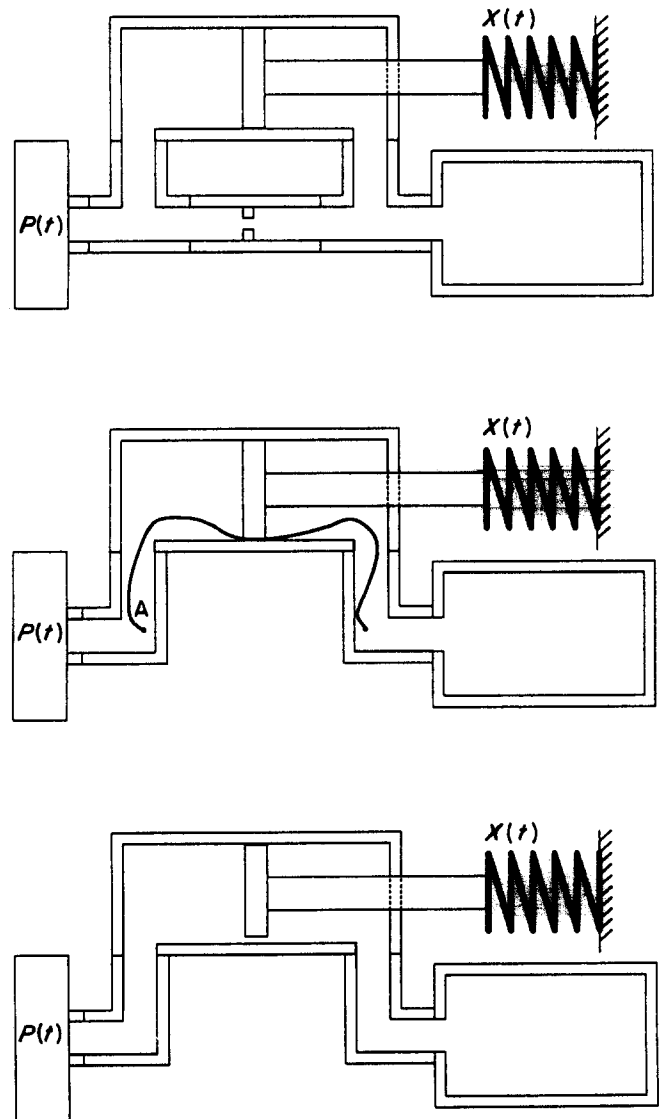


Figure 5. Eliminating the fluid resistance

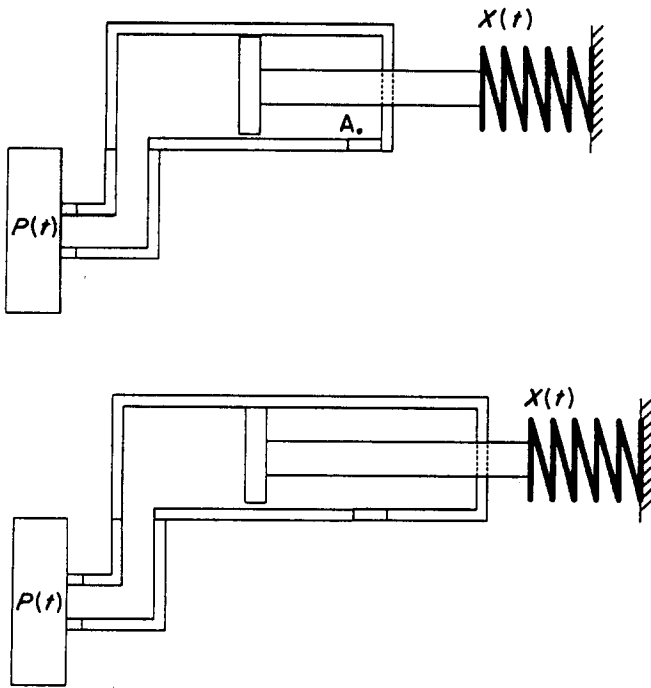


Figure 6. Eliminating the fluid capacitance

figures illustrate the simplification of a rate-of-pressure indicator. The device consists of a piston-cylinder, a fluid resistance and a fluid capacitance. The output of the piston cylinder is attached to a spring (labelled $X(t)$). The input of the device is attached to a pressure source (labelled $P(t)$). As $P(t)$ changes, a pressure drop is created across the fluid resistance. This pressure drop causes the piston to move, deflecting the spring. As $P(t)$ stops changing, the pressure across the fluid resistance goes to zero and the piston returns to its neutral position.

Each description in the figures corresponds to a different state during the simplification. First, the fluid resistance is deleted from the design. Next, a potentially resistive feature is found with respect to points A and B. In this case the feature is a path between A and B passing between adjacent edges. This feature is modified to create a clearance between the adjacent edges, such that resistance is established between A and B. In this step the structural element implementing fluid resistance has been eliminated by creating a leakage path past the piston.

Second, the fluid capacitance is deleted from the design and a feature that can provide fluid capacitance is found with respect to point A. In this case, it is simply a cavity adjacent to point A. Finally this feature is modified by extending the cylinder length, thereby providing the capacitance of the deleted element. In this step, the fluid capacitance is eliminated by exploiting the fact that the cylinder already has some capacitance.

Deleting a structural element

A physical description of a design is represented as a collection of structural elements. The first step in the function sharing procedure is to remove one of these

structural elements. Removal of a structural element from a design may cause some side effects. For example, removing a fluid element from a design may cause leakage. Removing a mechanical-translational element may cause parts of the design to become disconnected. Because of these side effects, the design must also be repaired after the deletion step. In the case of deleted fluid elements, repairs consist of changing the connectors to which the element was attached, or inserting a plug in cases where the element is not attached to connectors. In the case of deleted translational elements, the repairs consist of reattaching detached regions of the device. In addition to removing the element and repairing the design, the deletion step must establish reference points in the design with respect to which alternative features should be found. We omit the details of the procedures for making repairs and establishing new reference points (they are given in Reference 1).

Recognizing alternative features

After deleting the structural element, the next step is to find alternative features in the physical description that can potentially implement the function of the element. For clarity we assume that the deleted element has only one intended function (the multi-function case is treated in Reference 1).

We have approached the feature finding task as a physical property recognition problem—the identification of one of a set of known configurations of structural primitives that can be modified such that they approximate the relevant function. For example, fluid resistance can be derived from a narrow passage between two edges, a long narrow channel through a solid region, or an orifice in a plate. For each of these ways of implementing resistance there is a physical feature that could be potentially modified to achieve the resistive function. For example, a path between two reference points that passes between two adjacent but detached edges could be resistive if a clearance were established between the edges. A path between two reference points that is obstructed by a solid wall could be resistive if a hole were punched in the wall. These relations constitute the function sharing knowledge base. The relations for fluid resistance are shown in Figure 7. The relations for the other functions in the domain are similar.

The procedures for recognizing any particular physical feature are *ad hoc* and specific to our two-and-one-half dimensional geometrical representation. For example, finding a path between two points obstructed by a wall involves first finding all internal walls in the design, and then performing a search along section faces from each reference point to each side of the internal wall. For the most part these procedures are straightforward, if somewhat tedious, computational geometry problems.

Modifying alternative features

Once the potentially useful features are found, the final

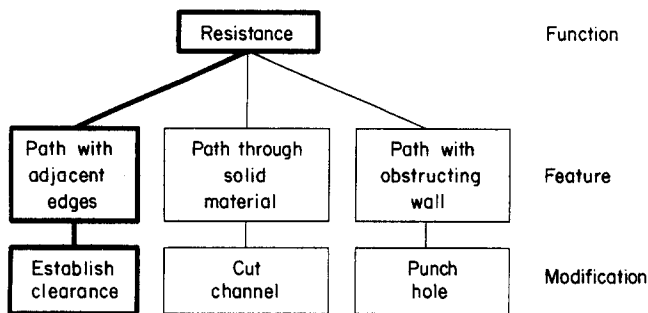


Figure 7. Organization of function sharing recognition and modification procedures

step is to execute the modification operators associated with each feature. In the case of the fluid resistance, for example, if the feature found by the recognition procedure were a path obstructed by a wall, the modification would be to punch a hole in the wall. Modifications are carried out only if the modification would not cause geometrical interference between different regions of the design. Geometrical interference is defined as the situation in which two sections of material overlap in the two-and-one-half dimensional space. The modification procedures, like the recognition procedures, are *ad hoc*. As an example, punching a hole in a section involves removing the existing section, replacing it with two new shorter sections with a gap between their ends, and revising the component connection list.

DISCUSSION OF FUNCTION SHARING PROCEDURE

The preceding sections have not dealt with many of the subtleties involved in function sharing. This section analyses the procedure from several perspectives and discusses some of the issues associated with adapting the procedure to other domains.

Meaning of physical descriptions

The physical descriptions produced by the function sharing procedure can be thought of as a way of assigning parameters to a design description. In the case of a piston-cylinder, there are many possible parameters that could be assigned to features in the physical description and that *may* be relevant to the design. The function sharing procedure is a way of assigning those parameters that should be considered in a particular design. For example, a piston-cylinder is normally parameterized by the piston area and the stroke length. There are, however, many other parameters that could relate to the element—among them the thermal conductivity of the cylinder wall, the size of the input and output ports, or the mass of the piston. After the function sharing procedure operated on the rate-of-pressure indicator example, two of these parameters were identified as important—the clearance between the piston and cylin-

der and the volume of one end of the cylinder. This identification resulted from the function sharing feature recognition and modification procedures. The procedure has identified a different and important set of parameters and the designer is alerted that the original design can be simplified if these parameters are considered when performing the detailed design and selecting dimensions.

Impairing existing functions

Modifying a design to exploit a secondary property of some structural element may invalidate the primary function of that element. For example, consider the result of eliminating the fluid capacitance in Figure 8. The expansion of the diameter of the resistance would impair its resistive function.

In our implementation of function sharing, all modifications that do not lead to geometrical interference are

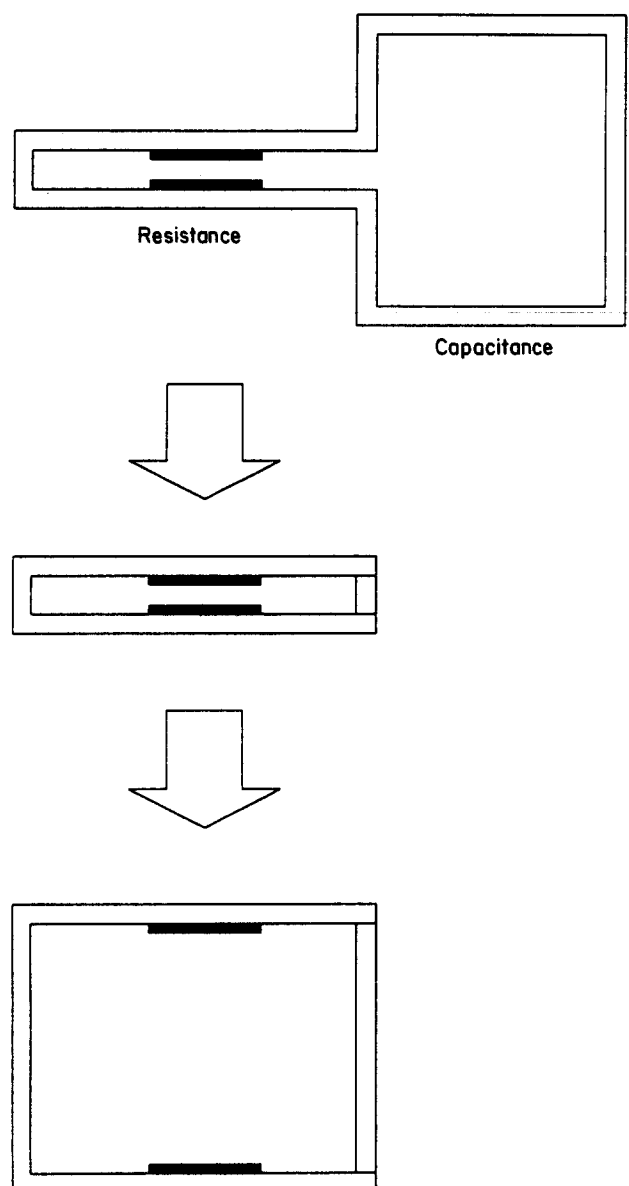


Figure 8. Modification invalidates primary function of element

carried out. We leave the judgement as to whether or not existing functions have been impaired to the designer. To do otherwise would require explicit knowledge of why a particular element functions the way it does, or would require analysis tools that can determine that a component no longer performs as specified. If the representation of each structural element also contained a set of constraints on allowable modifications to its structure, then some function-impairing modifications could also be avoided.

Design knowledge level

There are many alternative approaches to expressing function sharing design knowledge. We chose the physical feature level. This choice is a compromise between efficiency and expressiveness. For example, Figure 9 illustrates the variety of ways one might represent the concept of fluid resistance in a design system.

The simplest approach would be to identify particular

structural elements with particular functional elements. We call this the *component level*. This approach constrains the design system to use a particular component every time resistance is needed. Alternatively the system could represent resistance as a variety of generic physical features like holes, channels or porous materials. We call this the *feature level*. This approach gives the system more freedom to construct a resistive configuration. The system could also represent resistance as a mathematical concept with a generalized, parameterized model of geometry. We call this the *physics level*. Given this scheme, a system has a great deal of flexibility in synthesizing resistive configurations; however, with the flexibility and expressiveness come computational complexity. The reasoning required to derive a resistive configuration from equations at the physics level is severe; but finding resistance at the component level is trivial. We chose a middle ground. Representing resistance as a set of physical features gives flexibility and expressiveness, without swamping the system with complexity.

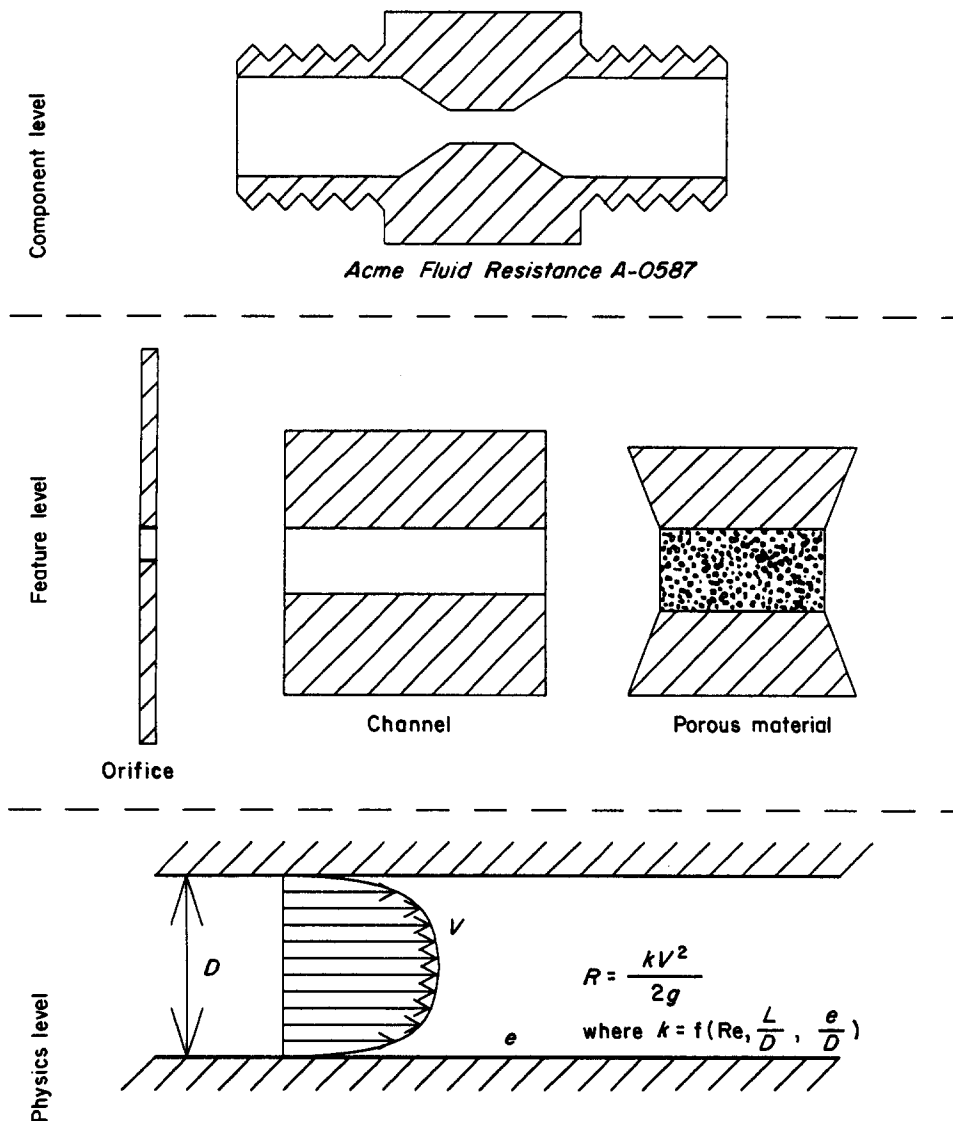


Figure 9. Knowledge level for fluid resistance

Example results from implementation

The ideas in this article have been implemented as a computer program. This subsection shows illustrations of the screen output of the program simplifying two different accelerometers. The accelerometer designs are highly stylized into our simple physical representation language. Each figure shows the progression of the design simplification as the user instructs the program to eliminate structural elements. Our implementation of the function sharing procedure leaves the control to the user. Specifically, the user selects an element to eliminate and chooses from among several modified designs. The control could also easily have been automatic, although this results in the generation of a collection of designs with a worst-case size that is exponential in the number of structural elements. In each of the examples only one or two of the outcomes of the program are shown, although in general there will be many designs generated. The particular examples shown in this section illustrate the range of simplification operations that the program can perform. There is a brief textual explanation for each figure.

Accelerometer 1

Figure 10 shows the program trace as it simplifies an accelerometer. An accelerometer is a device that produces a displacement (or other output quantity) proportional to the acceleration of the device reference frame. In most cases accelerometers will consist of some sort of mass and spring. The initial design in Figure 10 consists of a mass attached to a spring which is in turn attached to ground. The spring shown is the orthogonally-connected rectangular-prismatic version of an oval spring. The program is directed to eliminate the mass element. First the mass is deleted from the design. Next the system searches for alternative potentially massive features with respect to the top centre portion of the spring. The system finds the uppermost section of the spring has massive properties, and accentuates these properties by increasing the section thickness.

Accelerometer 2

Figure 11 shows the program operating on another accelerometer design. In this case, the initial design consists of a cantilever beam attached to a mass and damper. The beam and damper are attached to ground. The program is instructed to eliminate the mass. It first deletes the mass, and then looks for massive features with respect to the end of the cantilever beam. It finds that the beam itself is massive, and accentuates this property by increasing the beam thickness. Next, the program is instructed to eliminate the damper from the design. First it deletes the damper and then looks for alternative damping features with respect to the end of the cantilever beam. It finds that the beam has some damping because

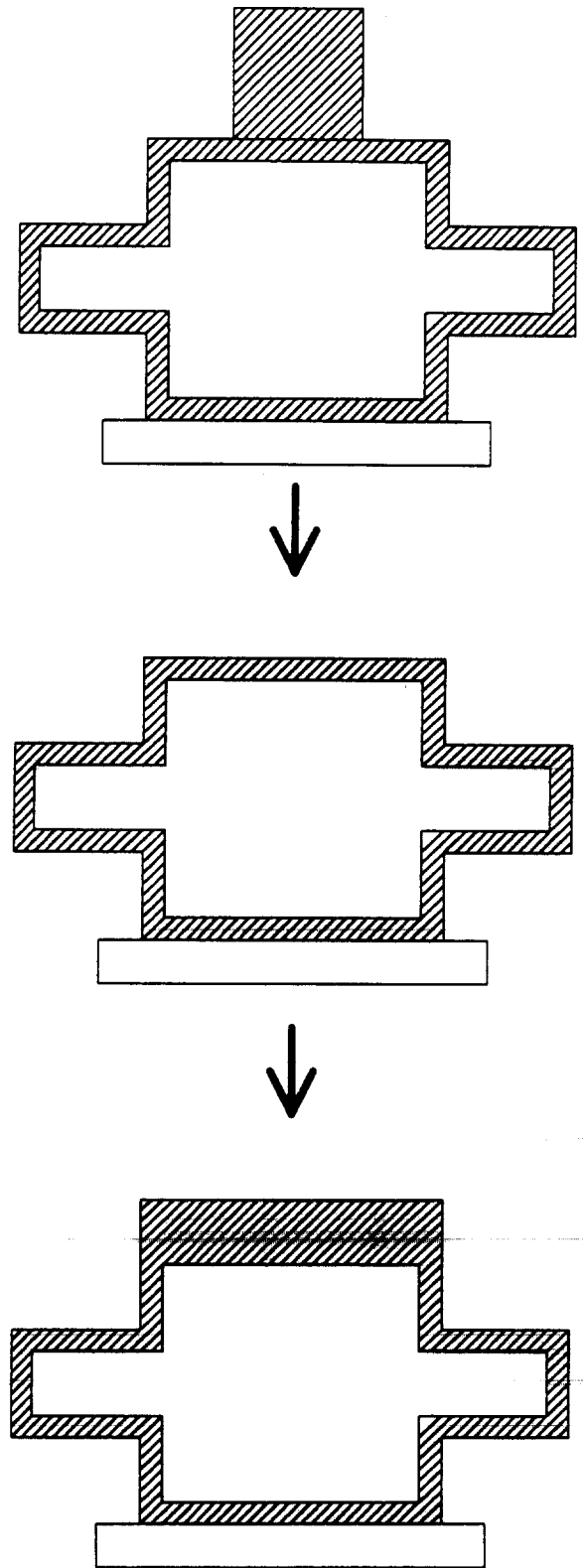


Figure 10. Accelerometer example 1

it moves in a gas environment causing drag. The program accentuates this property by changing the design environment to a viscous fluid (indicated by the dots). In another branch of the simplification, the program extends the piston thickness in order to replace the mass.

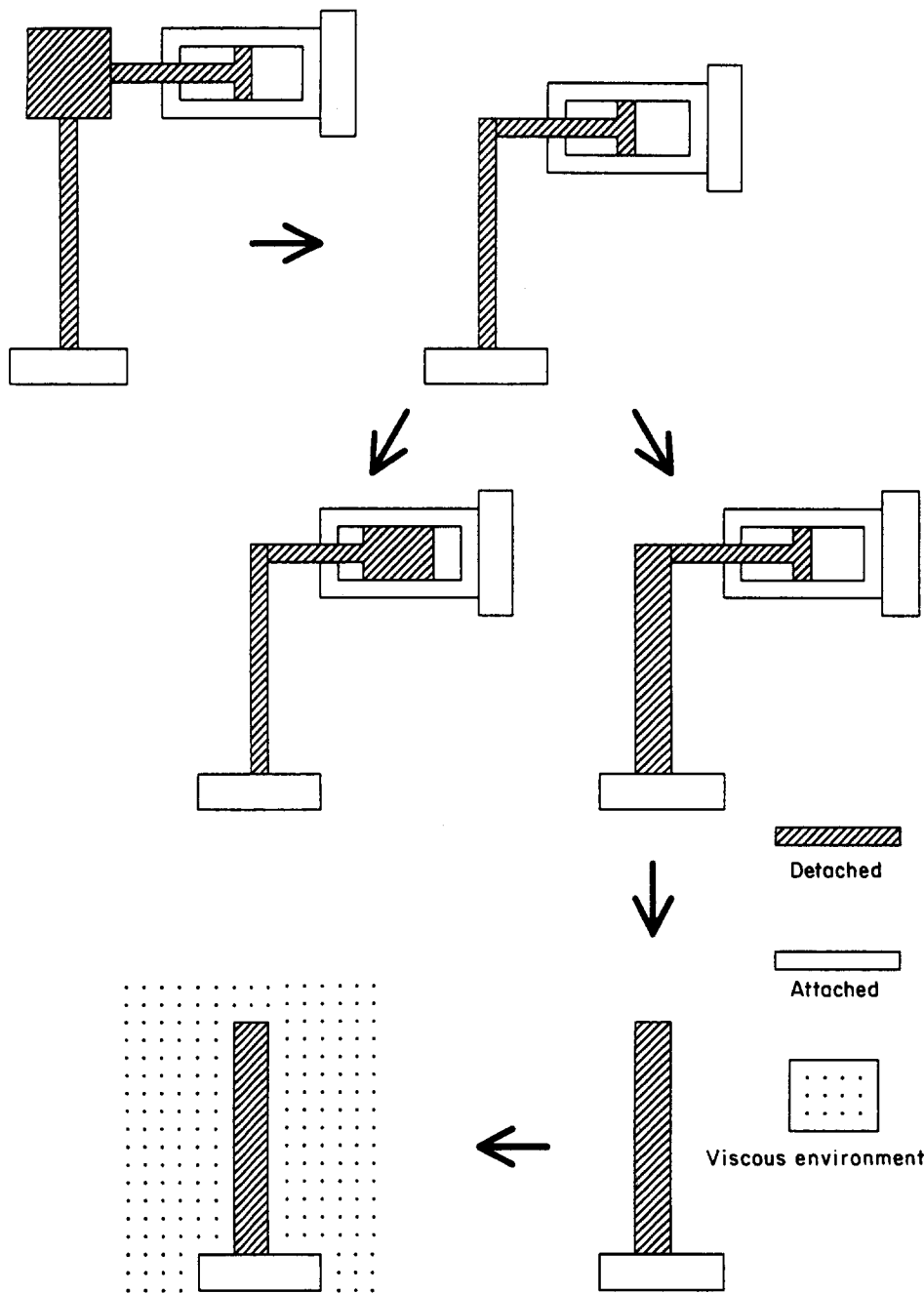


Figure 11. Accelerometer example 2

Novelty in mechanical design

A novel mechanical design can be defined as one that no one (or at least not the design community in question) has generated before. One hypothesis of this work is that the unbiased application of the physical-feature-based design operations in function sharing will yield some novel designs. One design the computer implementation generated truly surprised us. Figure 12 shows one branch of the function sharing procedure on the rate-of-climb example. In this case eliminating the fluid capacitance led to a novel use of the cavity associated with the space between the resistance and the piston. The recognition procedure identified the cavity between the resistance and the cylinder wall as potentially capacitive and the

modification procedure exploited the situation by punching an access hole in the cavity.

As long as the feature recognition procedures are sound, designs will be generated that are viable candidates. Some of these designs will be unanticipated because the recognition procedures are more thorough and faster than human designers, and because they do not encode any designer intention or functional information about the physical description.

Extensibility

Although this article describes only a first step towards understanding a complicated class of design activity, we

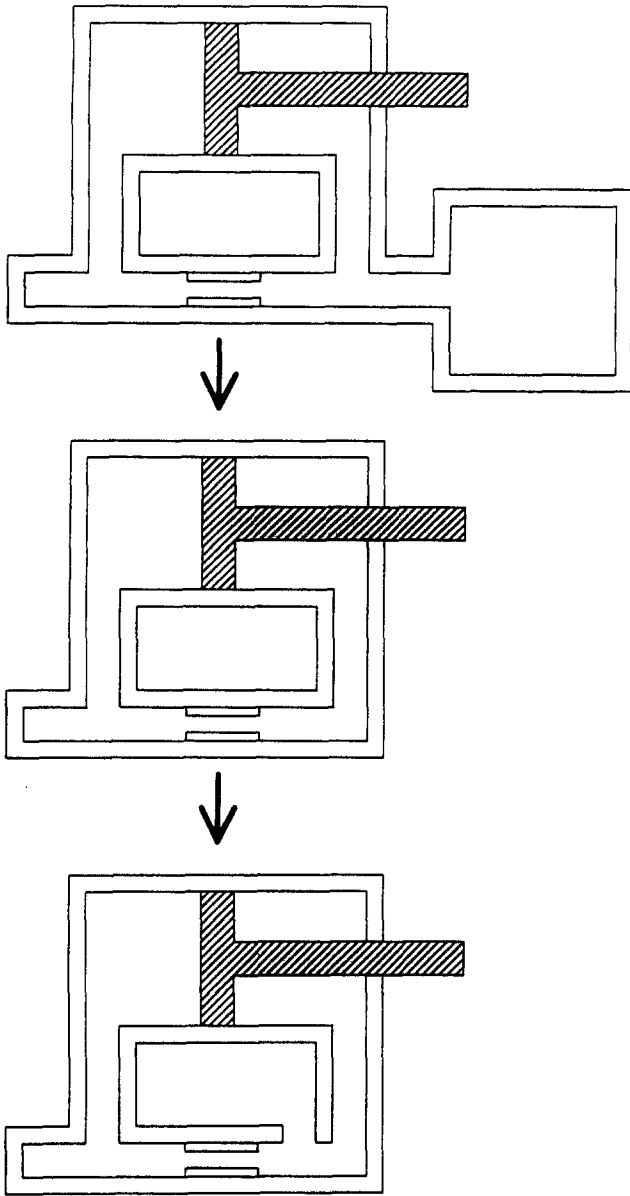


Figure 12. A surprising design

believe that the function sharing ideas presented in this article can be applied to other domains. There are several requirements on these domains, and on the way in which knowledge about the domains is organized. Certain properties of a domain also determine the success or failure of the function sharing procedure.

- There must be a definable language for describing the function of the constituent elements of a design. Designs in the dynamic systems domain can be described schematically with a concise functional language. Global device behaviour can be specified by a differential equation relating an input quantity to an output quantity. More important, a network of lumped-parameter elements (fluid resistances, fluid capacitances, etc.) can accurately describe the functional architecture of the device. The existence of a formal and concise language for the schematic description of the device makes the function sharing procedure possible: the function of any particular region of

the physical description can be precisely and completely specified, and there is a framework upon which the function sharing knowledge base can be built.

- There must be a definable mapping from function to structure. In the dynamic systems domain, each functional element corresponds to a set of physical features with which it can be implemented. Fluid resistance, for example, is a functional element that can be implemented with a hole in a plate, a gap between adjacent edges, or a channel cut through a solid. This mapping is accurate and concise. The corresponding mapping for *aesthetically pleasing* or even *aerodynamic* would be much more difficult to define. This is as much a requirement on the domain, as a requirement on the granularity of the functional language. In general the more fine-grained the functional language, the simpler the task of generating a function-feature mapping. An additional requirement in this vein is that the physical feature recognition problem must be computationally tractable.
- There must be locality of function in the physical design description. If the physical features associated with a particular functional element are highly distributed, then these features cannot be easily deleted from the design. In the dynamic systems domain this property is present, as indicated by the existence of definable reference points for a particular functional element.

Some domains that appear to meet these criteria are architectural floor plans, mechanical support and constraint (i.e., brackets, bearings and struts), chemical processes and machines for performing fixed automation tasks like assembly.

RELATED WORK

Although as far as we know no researchers have explicitly treated the problem of devising procedures for function sharing in design, several researchers have done work that relates in some way to the issues that we have addressed. We have built upon their ideas whenever possible. In this section, we give brief descriptions of publications relevant to our work from the artificial intelligence community, the design theory and methodology community, and the computer-aided design community.

Cagan² describes a method for deriving novel structural modifications to a beam design based on equations describing the design. The key insight is that mathematical integrals in the system description are indications that the structure of the design can be usefully decomposed into subregions. For example, since the strength of a bar is expressed in terms of an integral along the bar radius, it may make sense to consider the bar as consisting of concentric annuli. Within this formulation, an optimization of the design may specify zero density for the inner annuli, thus generating a tube.

Reference 3 is relevant to our work primarily because it introduces the no-function-in-structure principle. This principle states that systems intended to derive behaviour or function from a structural description of a device must represent the device with a description that is devoid of any functional information.

Freeman's paper⁴ is important because it is one of the first to discuss using connections between functional and structural descriptions as a knowledge base for performing design operations computationally.

Hirschtick⁵ deals with the problem of providing advice for the design of an aluminium extrusion cross-section. The basic approach is to use a hierarchical structure of features (for example, lines are bottom level features and knife-edges are top level features) as a knowledge base in order to recognize important design attributes in the two-dimensional cross-section. The recognized features serve as the input to a rule-based system that encodes design knowledge from an industrial aluminium extrusion design handbook.

Ishida⁶ deals with the problem of deriving behaviour from structure. The work focuses on a limited class of problematic behaviours (device leakage, for example). The objective of the research is to develop a statement of the unanticipated behaviour in terms of predicate operating on the geometrical representation of the problem. So device leakage, for example, is described as a property of the geometry of the device, expressed as a condition on the connectivity of a graph of part faces.

Jakiela⁷ integrates solids modelling with rules associated with design for manufacturability. In particular Jakiela has encoded the Boothroyd-Dewhurst design-for-assembly guidelines as procedures operating on a solid model description of a part. The system is interactive and suggests to the designer modifications that will make the designer modifications that will make the designed part easier to feed in an automatic bowl feeder. If the designer accepts the suggestions, the modifications are made automatically to the solid model of the part.

The program described by Murthy⁸ is based on the idea of first attempting to find a solution to a design problem in the parameter space associated with a given design configuration. If a solution cannot be found, then modification operators change the design configuration. This procedure was applied to the problem of designing beams for torsion and bending. In one case, the system generated a description of a tube by modifying a solid bar.

Reference 9 is one of the first papers attempting to describe the causal relationships within a mechanical device. This paper presents a language for describing devices, and develops a model in this language for a thermostat. The functional elements of the model are terms like: continuous and one-shot enablement, state coupling, state equivalence, state antagonism, threshold, and rate confluence. This language was designed to allow description of complex, highly nonlinear device behaviour that cannot be described with differential equations. Rieger uses these descriptions to simulate the behaviour of the devices by assigning a computational

behaviour to each functional element in the device description.

Rinderle¹⁰ discusses the relationship between form and function in design. He points out that in mechanical design there is often no clear hierarchical decomposition from function to form, as there is in VLSI design. He suggests that there are some fundamental relationships between form and function in mechanical design and that these relationships can be described. Rinderle postulates that the codification of some of these relationships could help novice designers make preliminary design decisions.

Suh¹¹ introduces the idea of design axioms. The paper presents the following hypothesis. *There exists a small set of global principles, or axioms, which can be applied to decisions made throughout the synthesis of a manufacturing system. These axioms constitute guidelines or decision rules which lead to 'correct' decisions, i.e., those which maximize the productivity of the total manufacturing system, in all cases.* The work described in this paper attempts to generate such a set of axioms. Two example axioms are: (1) minimize the number of functional requirements and constraints, (2) satisfy the primary functional requirement first. Satisfy the others in order of importance.

Sussman¹² introduces the idea of constraint nets as a computational structure. Rather than expressing mathematical relations as a unilateral procedure for computing certain values in terms of others, constraints express only the relationship between quantities. The constraint net can be used to compute a selected quantity from the remaining quantities. The other major idea in this paper is that of almost-hierarchical descriptions. This term is used to denote the fact that engineered devices have roughly hierarchical relations between functional descriptions and their structural implementations, but that this relation is sometimes muddled by function sharing in the design. Sussman introduces a concept called *slices* to deal with these almost-hierarchical descriptions.

ACKNOWLEDGEMENTS

This article describes work performed at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's research is provided in part by the Defense Advanced Research Project Agency of the United States Department of Defence under the Office of Naval Research contract N00014-85-K-0124, and the National Science Foundation under grant DMC 8618776.

REFERENCES

- 1 Ulrich, K T 'Computation and pre-parametric design' *Massachusetts Institute of Technology Artificial Intelligence Laboratory Technical Report 1043* (September 1988)
- 2 Cagan, J and Agogino, A M 'Innovative design of mechanical structures from first principles' *Intelligence Systems Research Group Working Paper 87-0801-2*, Department of Mechanical Engineering, University of California at Berkeley

- 3 **de Kleer, J and Brown, J S** 'A qualitative physics based on confluence' in Bobrow, D G (Ed) *Qualitative reasoning about physical systems*, MIT Press, Cambridge, MA, USA (1985)
- 4 **Freeman, P and Newell, A** 'A model for functional reasoning in design' *Proceedings of the Second International Joint Conference on Artificial Intelligence* (1971) pp 621–633
- 5 **Hirschtick, J** 'Geometric feature extraction using production rules' *S.M. thesis*, Massachusetts Institute of Technology, Department of Mechanical Engineering (1986)
- 6 **Ishida, T, Minowa, H and Nakajima, N** 'Detection of unanticipated functions of machines' *Proceedings of the International Symposium on Design and Synthesis*, Tokyo, Japan (July 1984) pp 21–26
- 7 **Jakiela, M J and Papalambros, P Y** 'Design and implementation of an intelligent CAD system' *Proceedings of the 1987 ASME Design Automation Conference*, Vol 1 pp 339–345
- 8 **Murthy, S S and Addanki, S** 'PROMPT: an innovative design tool' *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, WA (1987)
- 9 **Rieger, C and Grinberg, M** 'The declarative representation and procedural simulation of causality in physical mechanisms' *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, Vol 1 (1977) p 250
- 10 **Rinderle, J R** 'Function and form relationships: a basis for preliminary design' *Carnegie-Mellon University Engineering Design Research Center Report EDRC-24-05-87*, Pittsburgh, PA (1987)
- 11 **Suh, N P, Bell, A C and Gossard, D C** 'On an axiomatic approach to manufacturing and manufacturing systems' *Journal of Engineering for Industry* Vol 100 (May 1978) pp 127–130
- 12 **Sussman, G J and Steele Jr, G L** 'Constraints—a language for expressing almost-hierarchical descriptions' *Artificial Intelligence* Vol 14 (1980) pp 1–39