Proceedings of DETC99:
ASME Design Theory and Methodology Conference
September 12-15, 1999, Las Vegas, Nevada

DETC/DTM-8768

Transforming Functional Solutions into Physical Solutions

Y. C. Liu
Department of Engineering
Engineering Design Centre
University of Cambridge
Cambridge, UK
ycl27@eng.cam.ac.uk

A. Chakrabarti

T. P. Bligh

Keywords: ***Conceptual Design, Topological & Spatial Solution, Physical Solution, Structure-Behaviour-Function Relationships, Interface Compatibility***

ABSTRACT

The problem addressed is an issue of developing physical embodiments from a set of spatial configurations. These configurations are generated by a software program, FuncSION, for generating a wide range of concepts for mechanical design problems in conceptual design.

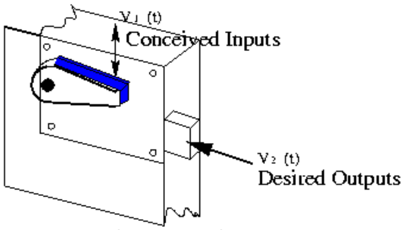
The method for transforming functional solutions to physical embodiments, consists of three steps: (1) to develop the relationships between each functional element and its physical embodiments, (2) to build the rule for ensuring interface compatibility between any two connecting objects, and (3) to develop reasoning procedures to replace each functional element in a spatial configuration with all its possible physical embodiments. Using this method, alternative physical embodiments for this spatial configuration can be found.

The outcome of the method is the presentation of physical embodiments, which leads to an improved visualisation of the spatial configurations, and an increase in the number of possible concepts.

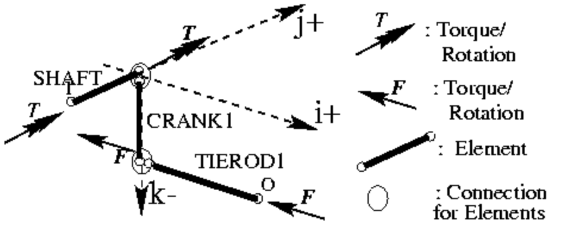
1 INTRODUCTION

It is widely believed that, in conceptual design, generating a wide range of concepts and allowing designers to explore and evaluate these, will increase the possibility of finding promising solutions (Adams, 1976; Ullman, 1992; Ulrich & Eppinger, 1995; Pahl & Beitz, 1996).

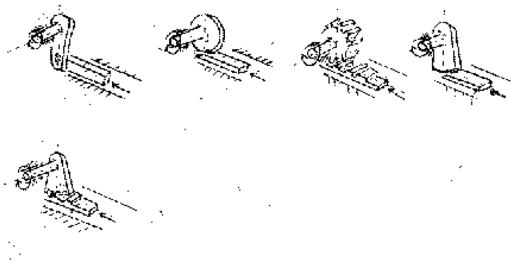
A project, in the Engineering Design Centre, Cambridge University, is dedicated to developing a program, FuncSION (which is an acronym for **F**unctional **S**ynthesizer for **I**nput **O**utput **N**etworks) [Chakrabarti & Bligh, 1994, 1996a, and 1996b], to produce a wide variety of ideas to mechanical design problems, which involve transmission and conversion of mechanical forces and motions. For a given design problem, the program can produce an exhaustive set of solution concepts, in terms of their topological and spatial configurations. These are then offered to the designers for exploration.



(a) Door latch problem representation



(b) One of the solutions generated by the program



(c) 5 possible embodiments of solution 1

Figure 1: Door latch problem, one of its possible solutions and some of its possible embodiments

FuncSION has been evaluated in a case study in terms of solution representation, novelty, and usefulness [Chakrabarti & Bligh, 1996c]. This showed that FuncSION generated a large number of ideas, many of which were not thought of by the designers, but were worth exploring in more detail. However, this evaluation also identified a number of issues to be solved, one of the most significant of which is that the representation of solutions is too abstract to easily understand. Solutions represented in FuncSION are visualized as abstract stick-like representations with spatial layout. Take a door latch design as an example (see Fig 1 (a)). If the latch input is a rotation, applied on the door, and the latch output a rectilinear motion, to enable disengagement of the latch from the doorframe, one of the generated concepts is shown in Fig 1 (b). This possible door latch solution is represented using a set of stick elements (called *functional elements*), and is a combination of an element (called *Shaft*) taking the input torque, transferring its output, a torque, to an element (called *Crank1*) that produces a translational output, to be transferred by an element (called *Tierod1*) to the desired output point. Possible physical embodiments of this spatial configuration are shown Fig 1 (c). If these spatial configurations could always be explained by means of physical representations as shown here, this would support the understanding of these abstract representations. Therefore, the problem addressed in this paper is an issue of developing physical embodiments from these spatial configurations.

1.1 DIFFICULTIES OF THIS RESEARCH

Developing the physical embodiments is hard by simply looking at the former abstract configurations. This is because solutions at the stick element level provide little information as to what physical objects and interfaces could be used to replace these spatial configurations. Difficulties in linking these two are:

7 Very little support theory is available, and thus the designer must rely on intuition and experience for transforming spatial into physical solutions. There is no general theory that relates function to object [Subramanian & Wang, 1995].

7 Each functional element can be represented by numerous physical objects. For example, in the door latch solution in Fig 1, the *Tierod1* functional element could be embodied using various physical objects which contain a translational input and output (see Fig 2). A reasoning procedure is necessary to link a functional element and its possible physical embodiments. However, if all possible geometric and dimensional variants of these are considered, the number of physical embodiments for the Tierod1 element could be infinite. How can we find a way to generalize geometry and dimensions so as to group them into a finite number of embodiments?

7 The connection between functional elements in a spatial configuration does not explicitly consider the characteristics of its interface. However, interfaces between connecting objects in mechanical designs can have various geometric forms and dimensions. It is a question of how to reason about suitable interfaces for two connecting objects.

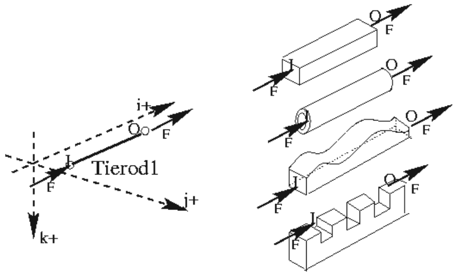


Figure 2: Possible physical solutions of a Tierod1 functional elemen

1.2 KEY IDEA

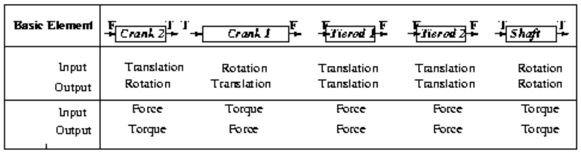
Previous research largely assumed that mechanism configurations are known and can be directly.utilized [Reuleaux, 1963; Kota & Chiou 1992; Prabhu & Taylor 1988 and 1989; Freudenstein & Maki, 1983; Hoeltzel et al., 1987; and Li et al., 1996]. Therefore, the behaviour and constraints of these mechanisms are pre-conceived and can be used for evaluation and selection. However, this approach lacks flexible combination of the mechanism elements, thereby, restricting the range of ideas that can be generated. Particularly, these approaches are weak for design when the system is still evolving and many of the functions are poorly understood.

Our method for transforming functional solution to physical embodiments, consists of three steps: (1) to develop the relationships between each functional element and its physical embodiments, (2) to build the rule for ensuring interface compatibility between any two connecting objects, and (3) to develop reasoning procedures to replace each functional element in a spatial configuration with all its possible physical embodiments. Using this method, alternative physical embodiments for this spatial configuration can be found.

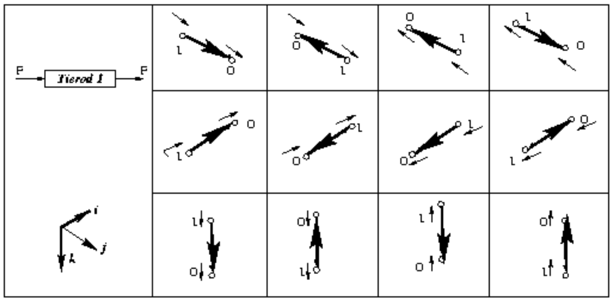
For a given spatial configuration, each of its functional elements is first replaced by a *generic* physical component or mechanism (component is defined in Section 3.1). These translated components or mechanisms can then be refined or modified to meet more function requirements for the continuity, linearity and reversibility of motion by qualifying their geometric forms and interfaces.

1.3 OUTLINE

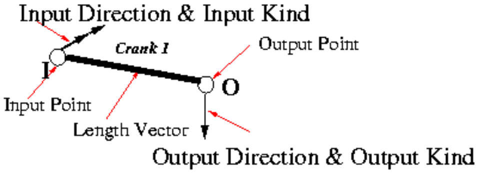
The rest of this paper is outlined as follows. We review the approach of FuncSION in Section 2, present our method in Section 3, investigate the method using a case study in Section 4, review the relevant work in Section 5, and finally present discussions and conclusions in Section 6.



(a) The set of elements used by FuncSION in synthesizing concepts



(b) A Tierod 1 with its 12 functional elements



(c) Structure of a Functional Element

Figure 3: Basic elements and functional elements

2 FUNCTIONAL SYNTHESIS APPROACH

FuncSION starts with the assumption that concepts can be generated by combining a set of *basic elements*. These elements are distilled from observing a wide variety of existing designs which were found to have common elements. For instance (see Fig 1(a)), the handle of a door latch is such that when pushed on the input end, it rotates at the other. In other words, it has a single input and output *kind*, one of which is a force (and translation) and the other is a torque (and rotation). This can be represented by a basic element called *Crank2*. In terms of input and output kind, *Crank2* has a translation to a rotation function or a force to a torque. Likewise, other basic elements with various motional purposes, were distilled. Fig 3(a) is part of the database of elements used by FuncSION for generating concepts.

The first step in FunSION is to synthesize a set of solutions which satisfy the input output kind requirements. These kind solutions are combination of basic elements, interconnected by certain kind, such that they transform the input to the output kind.

The second step is to generate spatial variations for these kind solutions. Each element has different orientations (see Fig 3(b), which shows that a *Tierod1* element has 12 orientations.), although they are functionally equivalent at the kind solution level. Each orientation describes how these elements can be oriented in space, and contains the relative position of the input and output. Each orientation is comprised of five parts: input kind (force or torque), input direction (+i, +j, +k, -i, -j, and -k), length vector, output kind and output direction, as shown in Fig 3(c). The length vector (called the *pin part*) is defined as a vector with a qualitative distance from the input point to the output point (each describes as the *dot part*, see Fig 3(c)).

The various orientations of each element are decided by the relationship between the input and output direction, and the length vector. In the handle of a door latch, for an instance, the input and output of this element in the spatial orientations have a definite relationship: they are orthogonal and non-intersecting.

The reasoning procedures generating spatial variations for the functional solutions are as follows. Each element has different orientations. Considering all possible orientations of each element in a solution, its alternative spatial configurations, which meet the rules of connection, are generated. Connections between these elements contain kinds, and directions of force or torque, and positions of possible components for the elements. For further details, see Chakrabarti and Bligh (1994, 1996a, 1996b, 1996c).

3 METHOD

Considering the transformation of a spatial configuration into its physical embodiments (see Fig 1), there are two questions to be answered: (1) what are the possible forms representing the pin part of the functional element? (2) What are the possible interfaces representing the dot part?

Considering (1), if we analyze the structural aspect of the pin part, two *generic attributes* are defined:

- Form: This is the abstract geometric representation of an object, and is the main attribute that transfers or transforms motions within itself. There are different geometric forms, such as plate, semi-disk, block, and rod.
- Support: This provides support that interacts with the structure's environment and sometimes contains the geometric coordinate of the object, such as the center of a circle. Common supports are revolute pair (turning pair), prismatic pair, screw pair, cylindric pair, spheric pair, and planar pair.

Considering (2), one generic attribute is defined:

- Interface: constitutes areas which objects contact each other. This attribute provides interactions between two connecting objects. There are at least two areas in each object: the input and output areas. Types of the interface are slot, tooth, joint, groove, plane, or traction.

If we analyze the structural aspect of a spur gear, the form is plate, the support is a revolute pair, and the meshing of the gear teeth is the interface. The interface of any two objects are similar to what Reuleaux (1963) called a *pair*.

Generic objects representing a functional element have a generic representation composed of form, interface, and support. Each object represents many *standard components*. For example, an object composed of a plate as the generic form, tooth as the interface, and a revolute pair as the support represents all sorts of gears. The relationship between generic objects and standard objects is shown in Fig. 4. Each standard object consists of its generic object added with its specific *descriptors*. Take a generic object with plate as form, tooth as interface and revolute pair as support, if its descriptors are circle, spur tooth, and bearing respectively. The standard object is a spur gear. If we change the *form descriptor* to rectangular and the remaining parts are the same, the resulting standard object is a rectangular spur gear.

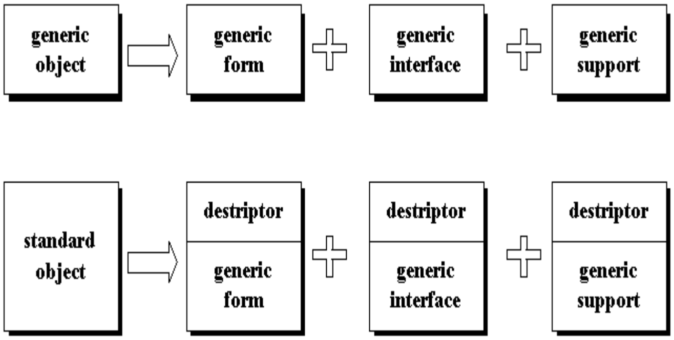


Figure 4: Representations of generic objects and standard objects

3.1 REASONING FOR PHYSICAL EMBODIMENTS

Before presenting the reasoning method, three terms, function, behaviour, and structure, are defined as follows:

- Function: is taken here as the intended behaviour and is viewed as a transformation between a set of input-output characteristics, such as 'to transmit rotary motion', 'to magnify input kind', or 'to convert rotary motion to rectilinear motion'.
- Structure: is defined as a component or assembly of components that describes geometric aspects of an object. Component is an individual geometric entity, such like a rod, a pin, a spring, or a shaft.
- Behaviour: is defined as `what a component or mechanism acturally does.

As far as each functional element is concerned, it contains information on function as well as on how possible objects are oriented. The problem is that it does not describe what these physical objects are. On the other hand, existing physical components contain geometric data without describing their possible functions and their abstract representations.

These two representations contain no direct link. One of the possible ways to deduce the relationship between them is to investigate the behaviour of some existing structures, analyze their behaviour in terms of functions, to build function-structure relationships. Additionally, by abstracting the representation of the existing structures, it is possible to transfer them into the functional element level. The reasoning idea, in summary, is: behaviour provides the linkage between function and its possible structures, and the abstraction of the structures transforms them to the functional element level. A method based on these Structure-Behaviour-Function (S-B-F) links, as well as on abstracting the level of structures, is therefore built.

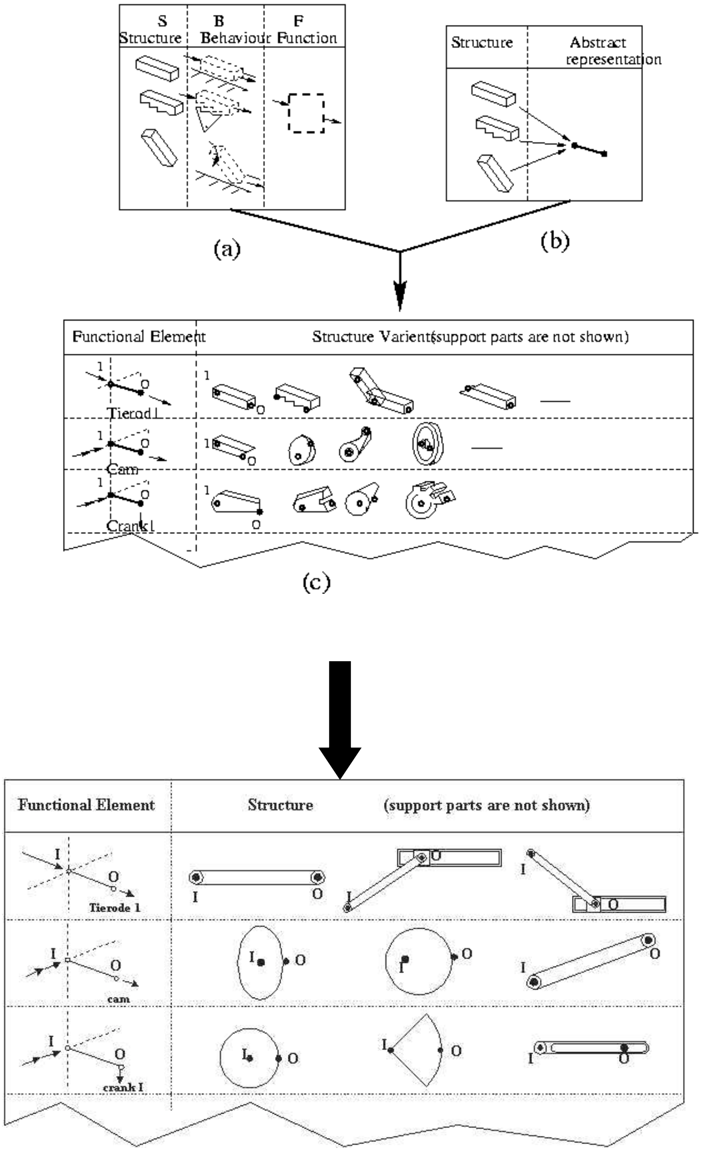
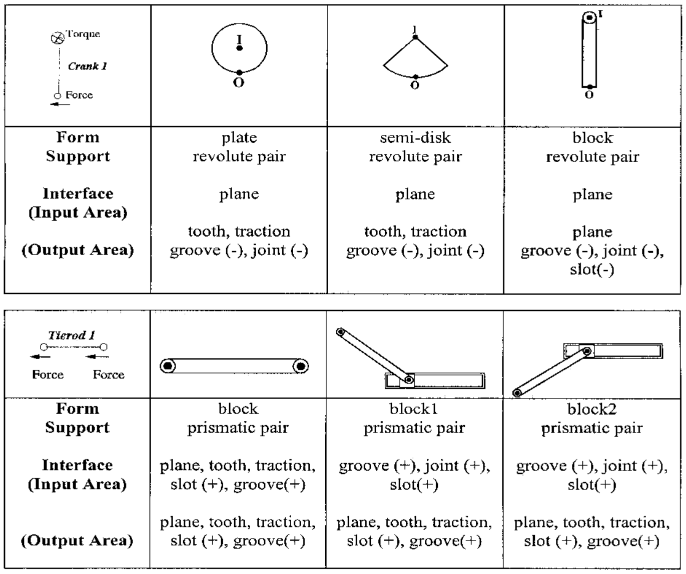


Figure 5: The method of building the relationship between functional element and their possible embodiments

Fig 5 shows a method for developing the links between physical structures and functional elements. By understanding the S-B-F relationships of many structures (Fig 5(a)), and converting the representation level of these structures into the functional element level (Fig 5(b)), the possible physical embodiments of any functional element can be derived (Fig 5(c)). These structures can then be classified into a set of generic representations with form, support, and interface (Fig 5(d)).

3.2 SOME RESULTS OF PHYSICAL REALIZATION

Fig 6 represents various structures for a *Crank1*, and *Tierod1*, which includes both components and mechanisms.



Form	plate	semi-disk	block
Support	revolute pair	revolute pair	revolute pair
Interface (Input Area)	plane	plane	plane
(Output Area)	tooth, traction groove (-), joint (-)	tooth, traction groove (-), joint (-)	plane groove (-), joint (-), slot(-)

Form	block	block1	block2
Support	prismatic pair	prismatic pair	prismatic pair
Interface (Input Area)	plane, tooth, traction, slot (+), groove(+)	groove (+), joint (+), slot(+)	groove (+), joint (+), slot(+)
(Output Area)	plane, tooth, traction, slot (+), groove(+)	plane, tooth, traction, slot (+), groove(+)	plane, tooth, traction, slot (+), groove(+)

Figure 6: Some functional elements and theirs possible components or mechanisms

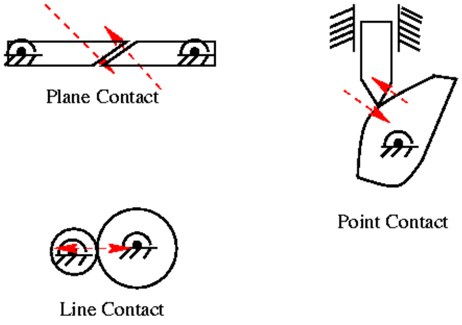
3.3 ENSURING COMPATIBILITY

Any solution composed of more than one functional elements has to ensure compatibility between its connecting objects in the solution. Both structural and behavioural aspects of objects contribute to the issue of compatibility. Analysis of these two aspects reveals that object compatibility contains (1) *configuration compatibility*, and (2) *interface compatibility*.

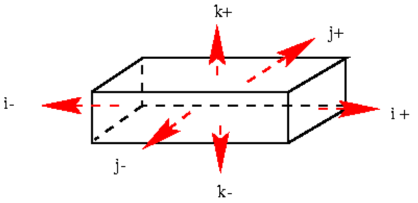
Configuration compatibility is that the space taken by one individual object can not be occupied by the other. However, each object in our method is qualitatively represented, the coordinate and dimension of the object are not quantified. Therefore, we suppose that any two connecting objects are located and dimensioned so as to meet configuration compatibility.

As far as the interface of two components is concerned, three important attributes of interfaces need to be considered.

- Normal vector: There are three types of contacting situations, see Fig 6 (a), namely, (1) *plane contact*, where the contacting area is a plane, (2) *line contact*, where the contacting area is a line, and (3) *point contact*, where the contacting area is a point. The common denominator of these situations is that the normal vectors of two contacting areas are coincident, with the exception of a point contact where the normal vector of the contacting point could be defined with all possible directions in space (see Fig 7 (a)). Any interfacing area is defined in terms of 6 normal vectors, oriented along i, j, k, -i, -j, or -k directions, (see Fig 7 (b)).



(a) Three types of contacting siturations



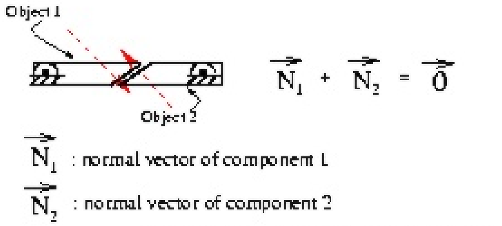
(b) 6 different normal vectors

Figure 7: Presentation of Normal Vectors

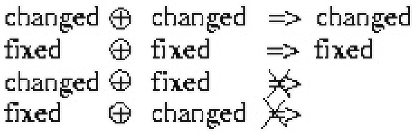
- Motion: the connection of two connecting objects permits a certain kind of motion. The motion of two connecting objects dictates the interface. For example, if the motion of the two objects is the same, their interface can be fixed. However, if the motion of one object is not always the same with that of the other, the connection has certain limits. The contacting area of a connection part can therefore be defined either *fixed* or *changeable*.
- Form: the interface of the contacting area of two objects must match.

The rule for ensuring interface compatibility is summarized as: two connecting objects must match in terms of their normal vector, motion, and interface at their contacting areas and is illustrated in Fig 8.

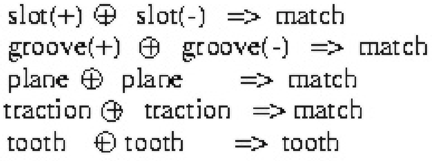
Fig 9 describes the matching process between two sets of objects, each set being alternative physical embodiments for a functional element. This process first selects an alternative from each set, followed by deciding whether the output area of the former object and the input area of the latter object meet the rule of interface match. This process has to be replaced for all lternative combinations.



(a) Normal Vector Matching



(b) Motion Matching



(c) Interface Matching

Figure 8: Interface Feasibility

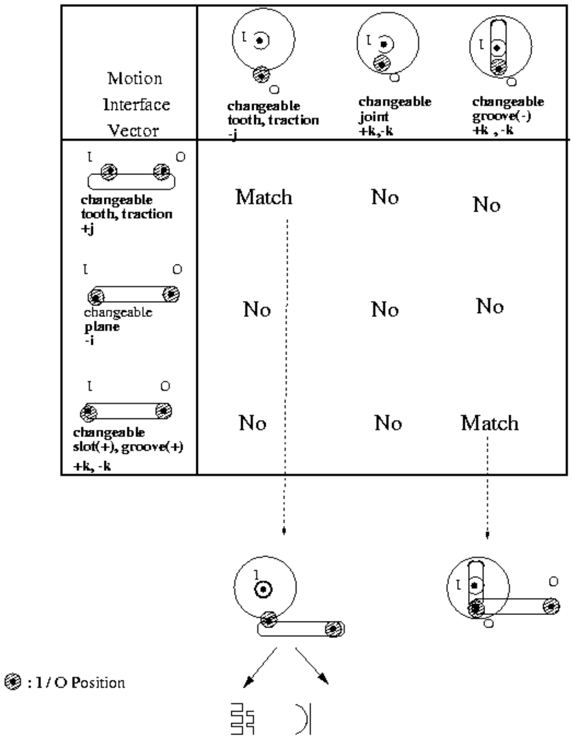


Figure 9: Interface compatibility matching for two objects

4 EXAMPLE

An example presented here demonstrates how to generate solution concepts. The design problem is to fulfil certain inputs and outputs, given a set of basic elements, by generating a set of physical solutions. A door latch problem with a rotational motion as input is chosen (see Figure 1(a)). This must be converted into a translational output motion to disengage the latch from the doorframe (this part of description is treated as *part I* of the problem). When we releave the input, the latch must move back to its original position (treated as *part II* of the problem).

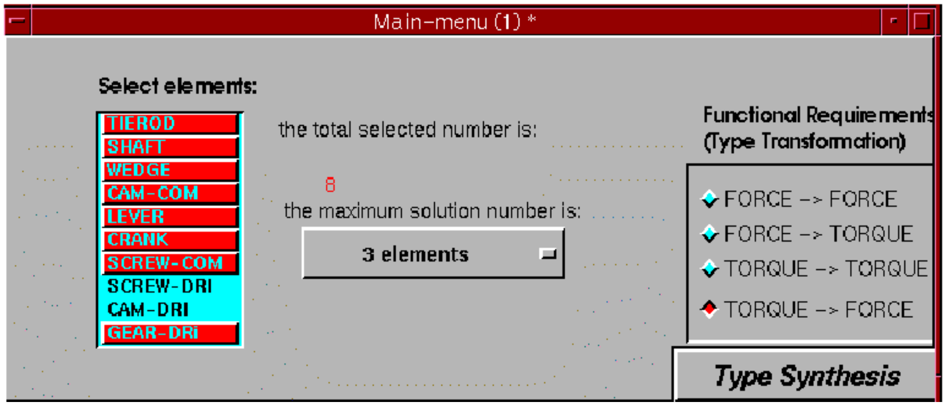
In our problem-solving strategy, part I of the problem is first considered to generate possible candidate solutions, followed by a modification of these solutions to meet part II. By considering part I, solutions are gradually detailed from functional descriptions to embodied physical descriptions, and solutions which meet all requirements emerge.

A set of possible concepts are generated first by means of kind synthesis, these concepts meet the input-output kind. Possible spatial configurations of these concepts are then generated by means of spatial synthesis; these configurations meet the requirement of I-O motion, motion direction, and relative position of input-output point. Possible physical embodiments are now generated, and these embodiments can be further detailed to meet the other requirements.

4.1 IMPLEMENTATION OF THE METHOD

The procedure for generating physical solutions is described below:

1. Choose possible basic elements that might be used in solving the design problem. In this example, eight basic elements are chosen (see Fig 10).
2. Specify the requirement in terms of given input and output kinds required at an instant of time. Fig 10 presents that (i) the requirements of the input and output kinds are: torque and force, (ii) all eight basic elements are selected for synthesis with the maximum allowable number, three, of basic elements per solution to be three. (Solutions generated by FuncSION are dependent on the selection of the number and the types of elements. In order not to generate too many solutions, the maximum allowable number of elements was limited to three.)
3. As a result of kind synthesis, 84 solutions are generated (see Fig 11).
4. Select any one of the solutions. For example, the solution with a Shaft, a Crank1, and a Tierod1 is selected.
5. Give directions of the input effort and output effort. The input direction and the output direction are: + j to - i.
6. Process spatial synthesis, the result of which is the generation of a number of topological & spatial solutions (see Fig 12).



Select elements:

- TIEROD
- SHAFT
- WEDGE
- CAM-COM
- LEVER
- CRANK
- SCREW-COM
- SCREW-DRI
- CAM-DRI
- GEAR-DRI

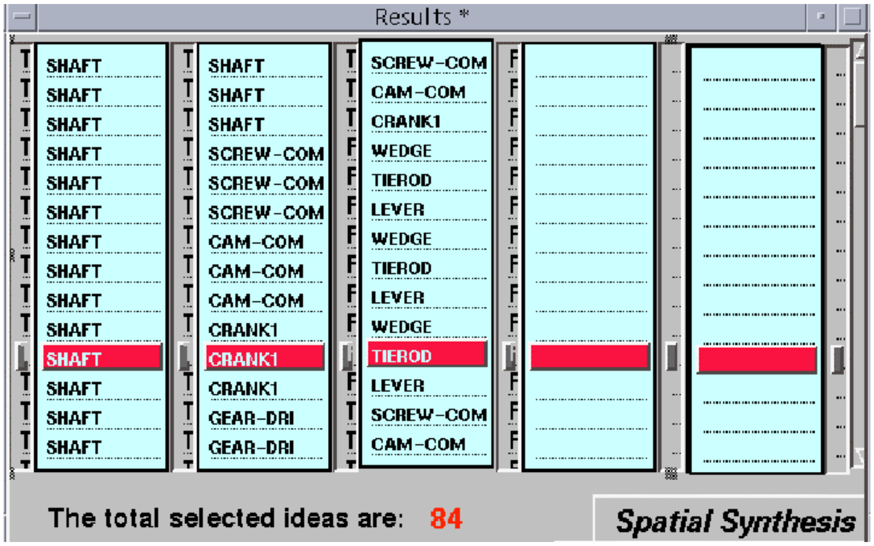
the total selected number is: 8

the maximum solution number is: 3 elements

Functional Requirements (Type Transformation)

- ◆ FORCE –> FORCE
- ◆ FORCE –> TORQUE
- ◆ TORQUE –> TORQUE
- ◆ TORQUE –> FORCE

Figure 10: The main menu of FuncSION



T	SHAFT	T	SHAFT	T	SCREW-COM	F
T	SHAFT	T	SHAFT	T	CAM-COM	F
T	SHAFT	T	SHAFT	T	CRANK1	F
T	SHAFT	T	SCREW-COM	F	WEDGE	F
T	SHAFT	T	SCREW-COM	F	TIEROD	F
T	SHAFT	T	SCREW-COM	F	LEVER	F
T	SHAFT	T	CAM-COM	F	WEDGE	F
T	SHAFT	T	CAM-COM	F	TIEROD	F
T	SHAFT	T	CAM-COM	F	LEVER	F
T	SHAFT	T	CRANK1	F	WEDGE	F
	SHAFT		CRANK1		TIEROD	
T	SHAFT	T	CRANK1	F	LEVER	F
T	SHAFT	T	GEAR-DRI	T	SCREW-COM	F
T	SHAFT	T	GEAR-DRI	T	CAM-COM	F

The total selected ideas are: 84

Figure 11: Parts of the results of kind synthesis

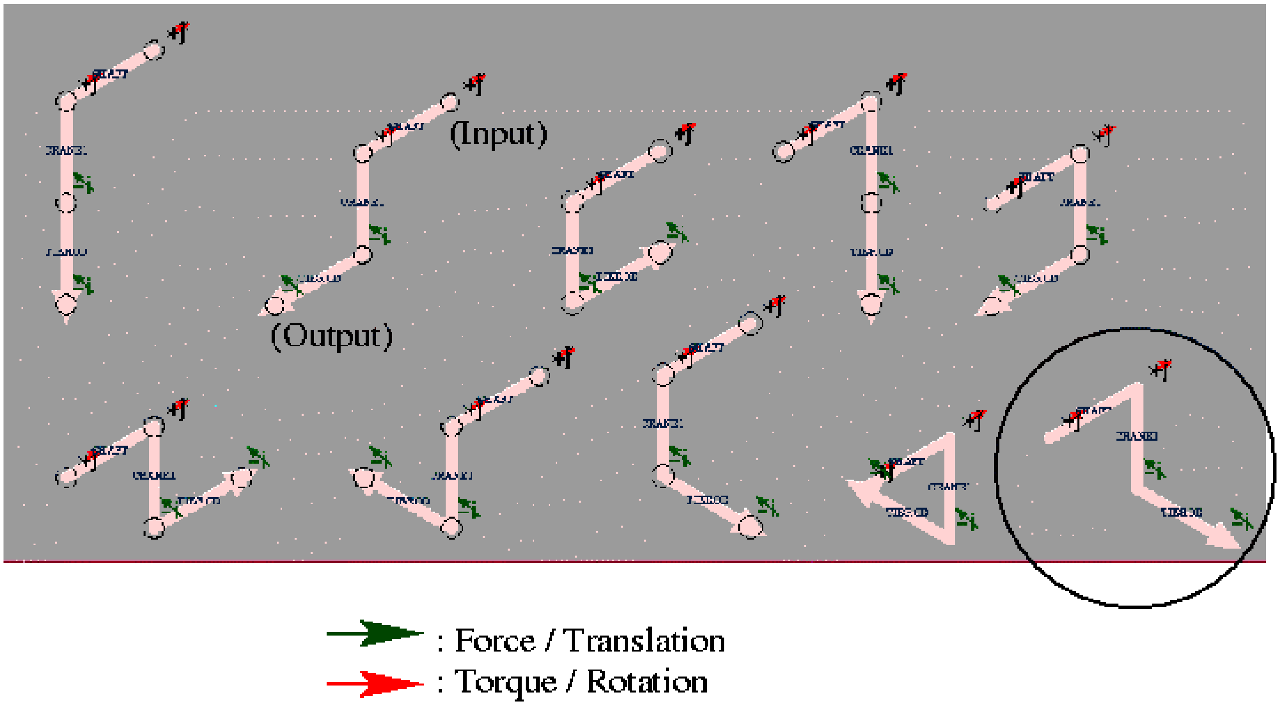


Figure 12: Parts of the results of spatial synthesis

7 Specify the relative position of input and output. For example, here, the relative position of the input and output is offset in i+, j+, and k- directions from the input point. Fig 12 shows that the chosen functional solution composed of a Shaft, a Crank1, and Tierod1 elements has ten spatial solutions. The circled solution which meets the requirement of the relative position is chosen for translation into physical embodiments.

8 Carry out physical translation. This step is to translate the abstract spatial configuration into its physical embodiments. The database of translated components with respect to each functional element are considered. Fig 13 shows some of the potential embodiments. Automating this step is still under development.

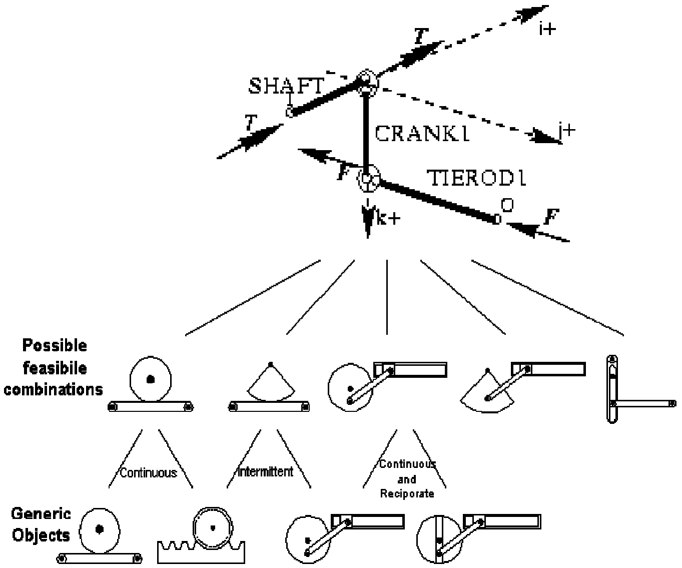


Figure 13: Possible physical embodiments of spatial configurations

4.2 MODIFICATION

Having generated the generic physical embodiments of the spatial solution in step 8, these can be detailed further. For example, if the form of a component is a plate, it can be further classified in terms of various form descriptors such as circular, elliptical, or rectangular. If the interface of a component is toothed, it can be classified in terms of spur, bevel, or worm geap types. If the support part of a component is a revolute pair, it can be further classified as a bearing. At this level, one can decide on the linearity of the input-output motion; for example, two circle spur gears make for linear motion, while two rectangular spur gears produce non-linear motion. Further classification of the generic physical embodiments is shown in Fig 14.

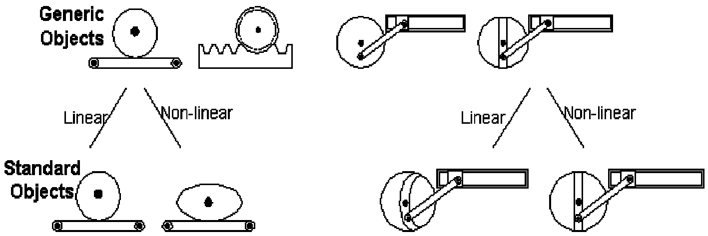


Figure 14: Further classification of the generic physical Embodiments

Having decided on the physical embodiments, part II of the problem can be considered. In order for the latch to return to the original position, an energy source must be considered and added to the design. One of the possible sources is the use of springs.

5 LITERAUTRE REVIEW

Hoover & Rinderle (1989) describe the nature of mechanical design problems, and discuss strategies for decomposition and transformation. They identify the difference between mechanical devices from other design principles, and recognise and exploit the physical and behavioural attributes (input-output type, ratio, motion direction, position, and number) of mechanical assemblies. Design requirements are represented quantitatively. Many researchers have explored the use of bond graphs as a design tool including Prabhu & Taylor (1988, and 1989), Finger & Rinderle (1989), Ulrich & Seering (1989), Rinderle & Balasubramaniam (1990). However, bond graphs provide limited guidance for transforming the function structure to a physical description of the device. The ability to represent form-related information such as shape, geometry and interface is absent. Ulrich & Seering (1989) outline a ‘design and debug’ strategy which generates an initial solution to meet input-output behavioural specifications and then debugs the solution to meet the full behavioural specifications. Prabhu & Taylor (1988 and 1989) represent solution concepts as a network (based on bond graphs) connecting various ‘ports’. Functional requirements contain scalar requirements - the 'values' of the power variables, and vector requirements. A design is generated to satisfy scalar requirements and thus is modified to satisfy vector requirements. Kota & Chiou (1992) propose a qualitative matrix representation for synthesis of mechanisms. They use mechanisms as building blocks of solution concepts. The functionality of each building block is expressed in a matrix style containing motion transformation and a sequence of constraint. Li et al. (1996) extend Hoover & Rinderle's method (1989) to a more general domain, mechanism design. The idea of their transformation rules is similar to Kota's method.

Fenves & Baker (1987) work on a spatial and functional representation language for structural designs. He uses operators that execute a known ‘grammar’ to generate architectural layouts as well as structural and functional configurations. Lai & Wilson (1987) have created a formal, English-language-base system called FDL for representing the function and structure of mechanical designs. While FDL can represent the function and form of a design, it provides no assistance in transforming a functional description into a physical description. In Pahl & Beitz's (1996) synthesis strategy, there is little guidance for transforming the function structure to a physical description of a device. A refinement of their approach has been done by Roth (1987) who uses multi-level representation models to provide a guide from abstract function structures to physical embodiments.

Welch & Dixon (1992) present a new representation - behaviour graphs which are based on bond graphs and qualitative physics. Behaviour graphs transform function requirement relationships to relationships based on physical principles and phenomena. The behaviour representation provides a way of systematically exploring a wide variety of solution principles without prejudice to particular artifacts, and is opposed to mapping directly from function to form. This viewpoint is similar to our transformation strategy. However, their possible embodiments which meet specified behaviour is limited to existing assemblies, and the consideration of possible solutions’ position and orientation is not exhaustive. Williams (1992) uses a graph-based representation of behaviour to capture the qualitative relationships between parameters. Navinchadra et al., (1991) also proposed representations of behaviour. Both their approaches are limited to behavioural descriptions at the parametric level. Gero et al. (1992) treate conceptual design as a transformation from function through behaviour to structure. They submit that behaviour serves as the platform form reasoning between the two, which is similar to us.

A few attempts have been cited in the literature about the use of configuration space for synthesis [Joskowicz & Addanki, 1988; Sun & Faltings, 1994] (these are often limited to interaction between two elements). However, their approaches seem more suitable for simulation than for synthesis. A refinement of Joskowicez & Addanki’s work is seen in Subramanian & Wang (1995) who present algorithms for kinematic synthesis of mechanisms from functional requirements which integrate methods in qualitative physics and constraint programming based on configuration spaces. As output, it produces a systematic enumeration of mechanism topologies and geometry that satisfy the given requirements. The position and orientations of mechanisms can be determined, and solutions are represented in terms of three-dimensional rigid parts.

In our approach, there are five key aspects:

1. Possible groups of solution concepts based on physical principles are generated by a set of basic elements (building blocks). The generation is based on composition rather than decomposition. This can exhaustively consider all possible groups of concepts which are likely to become design solutions, and would reduce the chance of missing promising ideas.
2. Each building block represents not only mechanisms, but also components. This expands the range of possible solution concepts and extends the use of this method from mechanism design to other applications, in contrast to many other approaches.
3. An intermediate representation level - spatial configuration is used to link the gap between function and its possible embodiments. All 3D spatial configurations are exhaustively generated for evaluation.
4. The interface between each physical object is explicitly considered; and is based on rules for ensuring the compatibility rather than only implicitly developed by designers.
5. Ullman (1993) indicates that studies of engineers show that the development of function is only possible with the parallel development of at least abstract forms. The function of sub systems, assembles, components and features evolve as decisions about the form of the product evolves. This approach attempts to support co-evolution of form and function.

6. CONCLUSIONS & DISCUSSIONS

This paper tackles one of the key issues in conceptual design - to transform functional solutions to their possible physical embodiments. The method proposed includes, (1) the knowledge of physical realizations for functional elements, and (2) the rule for ensuring interface compatibility between components. The outcome of the method is the generation of generic physical embodiments, which should lead to an improved visualization of spatial configurations, and an increase in the number of possible concepts. This approach at its present state should support the investigation of simple mechanical devices.

This approach is different from most of the current approaches which map functions into well-developed devices or mechanisms. This will enhance the opportunity for generating novel solutions. Although this approach seems to be able to generate a wider range of solutions, some aspects of the interface between successive components, such as the degree of freedom, and the form of the interface, need further investigation. Additionally, it becomes increasingly complicated when dealing with solutions which contain many basic elements.

Future work involves extending the set of building blocks, integrating mechanical objects with multi-domain elements, and considering multi-input multi-output requirements. A computer tool, based on this research, is under development.

REFERENCE

Adams, J.L., (1976), “Conceptual Blockbusting - A Guide to Better Ideas,” Norton, New York.

Chakrabarti, A., and Bligh, T.P., (1994), “An approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part I: Knowledge Representation,” *Research in Engineering Design*, Vol. 6, No. 3, pp. 127-141.

Chakrabarti, A., and Bligh, T.P., (1996a), “An approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part II: Kind Synthesis,” *Research in Engineering Design*, Vol. 8, No. 1, pp. 52-56.

Chakrabarti, A., and Bligh, T.P., (1996b), “An approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part III: Spatial Synthesis,” *Research in Engineering Design*, Vol. 8, No. 2, pp. 116-124.

Chakrabarti, A., and Bligh, T.P., (1996c), “An approach to Functional Synthesis of Mechanical Design Concepts: Theory, Application, and Emerging Research Issues,” *AI EDAM*, Vol. 10, pp. 313-331.

Fenves, S. & Baker, N. (1987), “Spatial and Functional Representation Language for Structural Design,” IFIP 5.2, North Holland, pp. 511-530.

Finger, S. and Rinderle, J., (1989), "A Transformational Approach to Mechanical Design Using a Bond Graph Grammar," Proceedings of the First ASME Design Theory and Methodology Conference, Montreal, Canada, September, pp. 107 – 116.

Freudenstein, F., and Maki, E.R., (1983), “Development of an Optimum Variable-Stroke-Internal-Combustion Engine Mechanism from the Viewpoint of Kinematic Structure,” *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 105, pp. 259-266.

Gero, J.S., Tham, K.W., and Lee, H.S., (1992), “Behaviour: A Link Between Function and Structure in Design,” Intelligent Computer Aided Design, D.C. Brown, M.Waldron and H. Yoshikawa (Editors), Elsevier Science Publishers B.V. (North-Holland), pp. 193-225, 1992.

Hoeltzel, D.A., Chieng, W.H., and Zissimides, J., (1987), “Knowledge Representation and Planning Control in an Expert System for the Creative Design of Mechanisms,” AI EDAM, Vol. 1, No. 2, pp. 119-137.

Hoover, S., and Rinderle, J., (1989), "A Synthesis Strategy for Mechanical Devices," *Research in Engineering Design*, Springer-Verlag, Vol. 1, No. 1, pp. 87- 103.

Joskowicz, L., and Addanki, S., (1988), "From Kinematics to Shape: An Approach to Innovative Design," Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88). Vol. 1, pp. 347-352.

Kota, S., and Chiou, S., (1992), "Conceptual Design of Mechanisms Based on Computational Synthesis and Simulation of Kinematic Building Block," *Research in Engineering Design*, Vol. 4, No. 2, pp. 365-372.

Lai, K., and Wilson, W.R.D., (1987), "FDL: A Language for Function Description and Rationalization in Mechanical Design," Computers in Engineering, ASME, New York, pp. 87-94.

Li, C., Tan, S., and, Chan, K., (1996), “A Qualitative and Heuristic Approach to the Conceptual Design of Mechanisms,” *Engineering Application of Artificial Intelligence*, Vol. 9, No. 1, pp. 17-31.

Navinchandra, D., Sycara, K., and Narasimhan, S., (1991), "A Transformational Approach to Case-Based Synthesis," *AI EDAM*, Vol. 5, No.1, pp. 31-45.

Pahl, G., and Beitz, W., (1996), “Engineering Design - A Systematic Approach,” Springer, London.

Roth, K., (1987), “Design Models and Design Catalogs,” ICED 87, Boston, August, pp.60-67.

Prabhu, D., and Taylor, D., (1988), “Some Issues in the Generation of the Topology of Systems with Constant Power Flow Input-Output Requirements,” Proceedings of ASME Design Automation Conference, pp. 41-48.

Prabhu, D., and Taylor, D., (1989), “Synthesis of Systems from Specifications Containing Orientations and Positions associated with Generalized Flow Variables,” Proceedings of ASME Design Automation Conference, pp. 273-280.

Reuleaux, F., (1963), “The Kinematics of Machinery: Outline of a Theory of Machines,” Dover Publications Inc., New York.

Rinderle, J., and Balasubramaniam, L., (1990), "Automated Modeling to Support Design," Proceedings of the Second ASME Design Theory and Methodology Conference, Chicago, IL, September, pp. 281-290.

Subramanian, D., and Wang, C.S., (1995), “Kinematic Synthesis with Configuration Spaces,” *Research in Engineering Design*, Vol. 7, pp. 193-213.

Sun, K., and Faltings, B., (1994), "Supporting Creative Mechanical Design," Proceedings of the Third International Conference on Artificial Intelligence in Design, pp. 39-56.

Ullman, D., (1992), “The Mechanical Design Process,” McGraw-Hill, New York.

Ullman, D., (1993), “A New View on Function Modeling,” ICED 93, The Hague, August, pp. 21-28.

Ulrich, K. T. and Eppinger, S. D., (1995), “Product Design and Development,” McGraw-Hill, New York.

Ulrich, K., and Seering, W., (1989), "Synthesis of Schematic Descriptions in Mechanical Design," *Research in Engineering Design*, Vol. 1, No. 1, pp. 3-18.

Welch, R. and Dixon, J., (1992) "Representing Function, Behavior and Structure During Conceptual Design," Proceedings of the Fourth ASME Design Theory and Methodology Conference, pp. 11-18.

Williams, B., (1992), "Interaction-based Invention: Designing Novel Devices from First Principles," Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), August, pp. 349-356.