The Knowledge Engineering Review, Vol 9:2, 1994, 163–199

A review of functionality modelling in design

JULIAN WINSOR
GIMT, Singapore

KENNETH MACCALLUM
CAD Centre, University of Strathclyde 75 Montrose Street Glasgow G1 1XJ, UK

Abstract

Recently there has been an increase in the number of computer aided design systems developed explicitly representing knowledge about the functionality of engineering designs. Reviewing these systems provides an understanding of the methods workers use to encapsulate knowledge of functionality within their systems. A number of issues are addressed to reveal the nature of their approaches. The developers' perception of functionality is discussed to identify variations in understanding of function and to establish the existence of any consensus. Methods of representing this knowledge are examined, thereby identifying representation types or combinations used and the advantages to be gained from any single representation. Illustrations of the manipulation of function shows how this type of knowledge can be used to support reasoning during early stage design. A survey of relationships with other design characteristics as a testimony to the manipulation of functionality is used to impact other aspects of a design. Through knowledge of relationships some models of the design process are posited by workers. A study of these bears evidence of the rôle of function in design and the stages at which its use is significant.

1 Introduction

For many years, modelling in computer aided design has been concerned with providing a description of the physical features and attributes of a design. Thus geometric modelling with related behaviour analysis has been a major contribution to the world of design.

A growing awareness of the opportunities offered by artificial intelligence (AI) techniques, and knowledge based systems in particular, opened up a completely new dimension for CAD researchers. A re-examination of design process theories was followed by a revision of views on relative rôles played by the computer and the designer in a CAD system, leading to concepts of an active agent supporting the designer in an intelligent way (MacCallum, 1987). Concepts of function and functionality are ubiquitous in design. Any textbook will emphasize that design exists to satisfy some needs, and that the process is therefore driven by an aim of specifying an artifact which will achieve some purpose or function. It will also make clear that as the process depends on being able to anticipate some future reality, it is necessary to be able to predict correctly how a design functions. Thus, function and functionality in the physical world has always been of central concern to the design engineer. Over the last two decades, a significant number of computer-aided design (CAD) system developers have recognized the importance of using this information and have been investigating ways to represent knowledge of the function and functionality of engineering designs within CAD systems.

This paper presents a critical review of the existing approaches taken to modelling with information of the functionality of engineering designs both with and without CAD system support. Work by authors (or groups of authors) is assessed within the context of a set of issues including; *perception of functionality, methods of representation, manipulating knowledge of functionality, relationships with other design characteristics* and *design process models*. Addressing these issues provides a format for a complete assessment of each piece of work in its own right, as well as a basis of comparison between work.

Initially, the perceptions of functionality are discussed. The differences form an important foundation for further studies of the subject. Methods used to represent knowledge of functionality are subsequently examined and a connection between representation and use is established. Work, including approaches to manipulating knowledge of functionality, is studied giving some insight into the breadth of concepts this type of knowledge can describe. In some cases, it is the intention to relate knowledge of functionality to other design primitive types, including behaviour and structure. Examples show how this has been tackled often through the use of frameworks relating functionality with other design parameters. A section focusing on models of the design process examines how knowledge of functionality can be used to assist the designer in the development of a model of a design.

Conclusions drawn from addressing each of the above issues are summarized and discussed. In support of this a comparative matrix (Table 4) presents the salient features of research in this field. Although it is desirable to include all of the relevant material in this field it is impractical. A number of key pieces of work have been selected to form the backbone of this review. The majority are examples taken from implemented CAD systems. Where appropriate, reference is made to other work approaching the use of functionality in general design work and not supported by CAD systems. Each of the key references are reviewed in detail, but taken individually they may not relate to all of the prescribed issues. However, as a collective they address all the major issues dealt with by the review.

The findings of the review help contribute to the development of a representational framework for describing a design's functionality and a model of the design process for manipulating the knowledge within the framework.

2 Scope and focus for the review

The coverage of this review can be delimited and characterized in a number of ways.

Firstly, as has already been indicated, a primary interest is in the contributions which can be made to supporting the design process. There is a large and very diverse literature which in some ways is relevant to the problems of encapsulating functionality in design. A bibliography of such literature is included as appendix to the review. In many cases this work derives from a more general research goal, perhaps associated with AI, or a need to represent function for some reason other than design. Only where work is of demonstrable relevance to design support is it included in this review. As an example, this paper chooses to include some work on qualitative physics because of its contribution to predicting and understanding how systems work, even though primary goals of this work have been to provide a generic reasoning capability in physics and to support a system diagnostic capability.

The recent increase in the number of researchers choosing to represent knowledge of functionality in CAD systems forms part of an attempt to use CAD systems during the conceptual stage of the design process. This is in line with the notion that design proceeds by manipulation of knowledge of the functionality of designs during the early stages of the design process, and that this manipulation is critical to the later stages of the design process (Pahl & Beitz, 1984; Roth, 1971).

It is natural in design research to seek commonality in methods and representations, even though there may be wide differences in the nature of the product produced from the process. This argument of generality versus specificity will continue. In reviewing material, however, there is a predominance of application oriented work leading to a belief that there are important characteristics in applications which constrain the approach adopted.

In this paper, we have drawn upon work from a broad range of application areas. Each field exerts its own set of influences upon the way functionality is represented and used. One of the more recent and most popular areas is mechanical engineering including Faltings (1991), Finger & Rinderle et al., (1989), Rodenacker (1976), Roth (1982), Sturges (1981) and Taura & Yoshikawa (1991). Much investigation has also been done in chemical engineering (Ulrich & Seering, 1989), control systems (Gawthrop et al., 1990, Gawthrop, 1990, Rasmussen 1984), electrical engineering (Mitchell et al., 1985), architecture (Eastman et al., 1989, Gero et al., 1991; Mitchell, 1989), process engineering (Elm, 1988; Forbus, 1984), and software engineering (Freeman & Newall, 1971). In contrast, some authors are attempting to create a universally applicable modelling methodology to encapsulate functionality (Goel, 1991; Goel & Chandrasekosan, 1989; Umeda, et al., 1990).

A final way of defining the scope for this review is in its organization. A set of principal issues are used to assess each piece of work and also to act as a common basis for making comparisons between the different workers. Listed below are these issues and associated sub-issues:

- *Perception of functionality*—for a general understanding of functionality, the variation of impressions of functionality and any consensus of opinion of its identity that may exist. The significance of perception is the ambiguity apparent when the term function is used. Understanding this ambiguity is fundamental to any kind of review.
- *Methods of representation*—to examine the variation in the formats used, useful combinations of representations, advantages and disadvantages of any one representation, and situations to which any one representation can be applied. Representation here is taken to be any type of formalism which describes concepts of function for the purpose of reasoning about function.
- *Manipulating knowledge of function*—illustrations of how this type of knowledge is used to support reasoning during the design process. This forms the reasoning part of a representation, but is separated from it to improve the classification structure for the review.
- *Relationships with design characteristics*—indications of how system developers see that the manipulation of functionality knowledge affects other aspects of a design model, and *vice versa*. This aspect is central to relating function and functionality to design. The ambiguity in the concept reappears and can be resolved through its position with respect to other parts of a design system.
- *Design process models*—to show the use of functionality knowledge as a support during the design process and the stages during the design process when functionality knowledge is significant. This last aspect is a final link to establishing how the ideas previously presented are incorporated within a process.

The principal issues do in themselves represent some kind of taxonomy for the subject and are used finally in the paper as a basis for comparing approaches.

3 Perception of functionality

The word "function" has been used in the literature either to indicate the *purpose* or the *action* of a design. Work addressing the issue of this ambiguity is Finger and Rinderle (1989b):

> "Mechanical engineers tend to use the words function and behaviour interchangeably. Qualitative physicists make a distinction between these words; that is, the design's *function* is what it is used for, while its *behaviour* is what it does."

The source of this confusion can be traced back to common usage, as defined in some principal dictionaries. Function is:

> "Mode of action or activity by which a thing fulfils its purpose". (*Oxford English*) (Fowler & Farber, 1984).

Alternatively;

> "An activity appropriate to any person or thing". (*Chambers 20th Century*) (Davidson et al., 1985)

Longman Synonyms (Urdang, 1986) provides the following for the usage of function as a noun; use, purpose and role. For function as a verb it suggests; to act, to work, to operate and to perform. In this last definition we recognize the use of function as a noun and a verb. For example, the function of a radiator is to give out heat into the air. But, it functions (verb) by passing electricity through a thin conductor wire wound round an insulator. In cases such as this it is easy to see that as a noun, function conveys a sense of purpose, while as a verb it conveys a sense of how it works. In summary, we conclude that *function* can be used in its noun sense, that is "the design's purpose", or it can be used as a verb, "the design's action". The principal difference between the usage as a verb and as a noun is that the verb gives a sense of the internal character of a design, that is, its action. In contrast, the noun usage presents information of the relationship between the design and the environment in which it operates; its purpose.

In the remainder of this section, we look at examples of work which view functionality as either purpose or action. Subsequent to this, other examples are discussed which offer an alternative viewpoint. The variation in interpretations will recur when we examine function as part of design and process representations.

3.1 Function as purpose

Eastman et al. (1989) unambiguously state, "By function we mean the purpose which the product addresses." In support of this view, Simon (1982) discusses the function of the artifact by referring to purpose. More specifically, Chakrabarti et al., (1992) view function as the purpose of the solution concept.

Jens Rasmussen (1984) also argues that the perception of functionality is subjective and personal in his work entitled, "On Information Processing and Human Machine Interaction—An Approach to Cognitive Engineering". Here the author studies abstraction hierarchies with particular reference to the control of systems. Rasmussen puts forward a clear statement about a designer's view of functionality; "The way in which functional properties of a system are perceived by a decision maker depends very much on the goals and intentions of the person." Umeda et al., (1990) also regard function as a subjective entity. Function to them depends upon the eventual use of the system. "A description of behaviour abstracted by human through recognition of the behaviour in order to use it." Rodenacker (1976) regards function and behaviour as interchangeable.

Gero et al. extends the understanding of function to "The design intentions or purpose (these are socially as well as technically developed)." In their appraisal of behaviour they see *functions* revealing intentions and *behaviour* spelling out how the artifact achieves functionality. Additionally they state that functions are labels of an artifact's purposes and form an important description, yet they seldom explicitly form an evaluable basis for analysis.

Here we see an important aspect of the concept of function as purpose emerging—the reference to behaviour. Behaviour, in this context, refers to external behaviour rather than some internal working. It becomes clear that for some kinds of artifacts there is an apparent identity between external behaviour and its purpose. Typically, these artifacts are associated with process systems where the function is to obtain some output in exchange for some input (radiator, for example). For other types of artifacts, however, it is more difficult to relate external behaviour to function (paperweight, for example).

In this classification, we restrict function to its purpose, even though that purpose may be to produce some output.

3.2 Function as action

In contrast to regarding function in its noun sense, a significant number of engineers use function in its verb sense as the *action* exhibited by an engineering design. Behaviour will appear again, but here we are specifically interested in the interpretation of function as behaviour.

Mitchell (1989) provides a very thorough discussion of the nature and use of functionality in design (more specifically architectural design). At the onset of the discussion he clearly states:

> "Essentially a description of form tells what an object *is*, while a functional description tells what it *accomplishes*. That is, specifies actions with effects that interest us."

Elm (1988) has developed the Function Structure Methodology (FSM) for the design of power station control systems for Westinghouse. Based on Rasmussen (1984), the work requires the modelling of systems in terms of its operational processes. He has chosen to define functionality as "A collection of plant processes which are controlled in a co-ordinated manner to impact an individual plant parameter in support of other plant processes." Ulrich & Seering (1989) have a similar approach to that of Elm. Although they do not explicitly provide a definition of function, they use *functional elements* in their work labelled according to the processes they provide e.g. resistance, capacitance, etc. These viewpoints are consistent with regarding functionality as a system's "mode of action". That is, considering a system's processes is equivalent to considering its "mode of action".

Another interpretation for functionality comes from Faltings (1991). He mentions, "note that I use the word *function* to mean something distinct from the *purpose* of the device." He provides an example of specifications for the design of a ratchet. One of these examples includes, "the permissible backlash in the clockwise direction is at most 0.25 rotations". Here he describes the required function as behaviour, "the backlash", or in other words as an action of the design. Importantly, he stresses that the definition of function should be made in the context of any restrictions imposed by the device's environment in which it operates. This is similar to an argument found in Simon (1982), stating functionality only holds for a specified environmental context. He therefore defines function as a pair (environment and behaviour).

Chandrasekaran (1989) describes functionality as, "what we want the artifact to do or to be." He gives further explanation by saying that functions can be expressed as a state or a series of desired states as in Umeda et al. (1990). He also adds that functions can be regarded as constraints and are special because they act as constraints on the behaviour of the device.

In summary, we see two distinct interpretations for function. The first is associated with the purpose of a design, with a possible link to external behaviour. The second sees function in terms of "action" aligning it with the "process" or "mode of action" of the design. As far as possible in this paper, unless we are referring to specific pieces of work we will use the term function as purpose, and functioning or functionality as mode of action.

4 Representation of functionality in design models

Representations of function and functionality have been classified into one of a set of general groups. The groups include: function grammars, input-output systems, qualitative physics and Bond Graphs. Each is reviewed in turn, and mention is made of the justification for using that particular representation. Where relevant it is intended that this section provides a comparison of the scope of usage for each representation type, illustrations of combinations of representations, and a summary of the main advantages and disadvantages of each type of representation.

4.1 Function grammars

The basis for any grammar for function derives from a formalization of common usage, typically expressed as a verb-noun pair. Pahl & Beitz (1984) give examples which include "increase pressure", "transfer torque" and "reduce speed". These examples can be seen to derive from an expression of manipulation of energy, material or information. The verb represents the action of the system on the noun, and the noun represents the energy, signals or matter handled by the system.

Another example of the verb-noun representation is Sturges (1981) and Sturges et al. (1991). They see function as "what a thing does"; they support their choice by saying the representation provides a common language across domains and it motivates the generation of alternatives. However, the authors warn that care should be taken when selecting nouns which may tend to obscure alternative design approaches or restrict creativity. In Lai (1987), the Functional Description Language uses verb-nouns to denote contact between components to assist in the design of artifacts for ease in assembly, an example is; base-support-bracket. The author's justification for such a representation is that it allows a simple qualitative comparison between structures that will eventually be spatially adjacent to one another.

It is clear that the application of verb-noun pairs as a type of function grammar is particularly related to the interpretation of function as purpose. As such, however, it is a weak representation, useful for description but with limited semantic content or reasoning potential. Its main application is in symbol level manipulation in a rule based or other pattern matching system.

A paper by Hundal (1990) denotes function by creating labels; examples are "sense", "amplify" and "convey". As an extension to this the authors have developed a classification for these basic function types, including store/supply, connect, branch, channel, change magnitude and convert. Further explanation of such a classification is presented in section 4.2.

A more formal approach can be developed if a suitable theoretical basis for representation can be defined. In the paper "CAD Framework Guided by General Design Theory" (Yoshikawa,1983) the author describes the need for design theory in a construction of CAD systems. The General Design Theory (GDT) provides a view for information about the structure of data in the designer's memory and the procedures by which a designer describes specifications. Within this Yoshikawa believes that the computer could overcome the problems of the finite capacity of human memory. He sees this as the cause of the designer's inability to consistently define the design's requirements in terms of feasible functions. He posits that a systematic and standardized description method of specification was required. Furthermore in this language the following must hold:

1. A meaning of a word is a function.
2. Each word must have a meaning as independent as possible of other words.
3. The language system must have syntactical rules: such as checking contradictory use of words and examining the completion of whole sentences.

Yoshikawa qualifies his ideas by stating that the language would be suitable only for a prespecified domain. If a general language were to be proposed then an intensive study of the function or the functional concept set would be required.

4.2 Function represented as input/output

It is common to find "function" represented in terms of the relationship between a system's inputs and outputs. These systems are regarded as having an element of flow, where an input traverses the system eventually to become its output (assuming there is no permanent storage). This flow may be the movement of information, matter and energy. Early examples may be found in Pahl & Beitz (1984) and Roth (1971, 1982). This representation is favoured by many CAD system developers. Describing the design in these terms means that attention can focus on it's internal character or mode of action. An example is Westinghouse's "Function Structure Methodology" (Elm, 1988). This makes use of input-output relations for the definition for function in work on the description of nuclear reactors with material (nuclear fuel) inputs to a nuclear driven power generator.

Various types of function can be differentiated in terms of their input-output characteristics. An example of this is the language in Goel, 1991. Here, selected functions express the manner in which the system's operation affects the state of a substance flowing through the system. The alterations of state include *transformation,* and *maintenance* (there is also the notion of state prevention). Elm (1988), Hundal (1988) and Imamura et al. (1988) also use similar functional description languages.

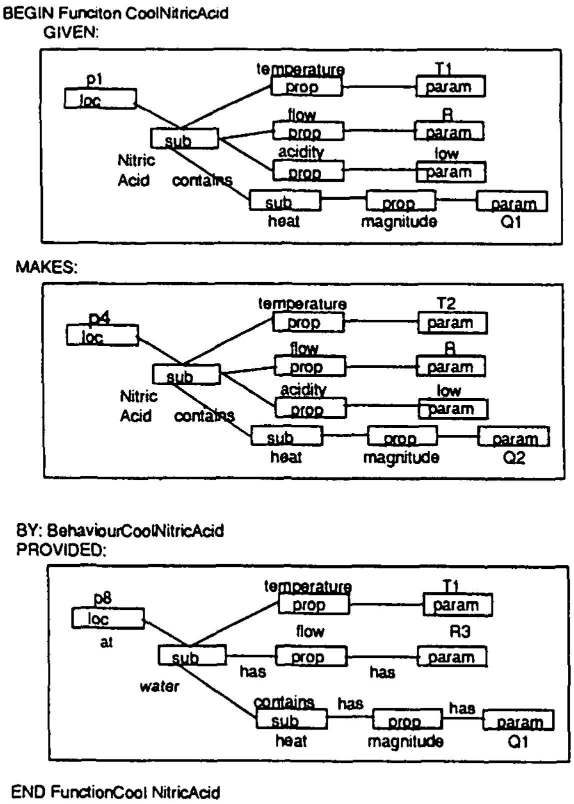


Figure 1 CoolNitricAcid Function Delivered by Nitric Acid Cooler (Goel, 1991)

The idea of change of states in Goel (1991) (Figure 1) is based on similar principles to Umeda et al's.(1990) concept of behaviour. Through the use of schemas the knowledge about a device's function is graphically represented. In Goel's case, a schema includes a label indicating the system's function. In addition, there is a GIVEN slot specifying the required behavioural state input and a MAKES slot that indicates the behavioural state output. In an example, the function is "CoolNitricAcid", taking an input through port $p1$ of nitric acid of temperature T_1, flowrate R, containing heat Q_1. If there is an input to port $p8$ of water with temperature T_1, flowrate $R3$ and containing heat Q_1 then the nitric acid output from port $p4$ will have temperature T_2, flowrate R and contain heat Q_2.

In his representation of function, Goel involves two issues including the content of the function and a language for representing the content of the function. Firstly, content of the function is based upon a theory of content that gives light to distinctions between functions. The example presented in the paper is of a designer thinking the function of a wire in an electrical circuit is to allow the movement of electricity, and that the function of a light bulb in the same circuit is to create light when it has an electrical supply. He mentions that such statements of functionality allow different types of inferences to be drawn which are dependent upon the task to be addressed. Essential to this is Goel's belief that if a system is to deliver a function it must be placed in a context of operation, that is it requires a supply of electricity. Furthermore, for a designer to properly draw functional inferences from a given situation he/she must have knowledge of the required task or purpose of the system.

Other authors differ in the way they relate function to the input-output characteristic of the system. In Chakrabarti (1989) a design is recognized as having an *input function* and an *output function*. Also, the relationship between the input and the output functions produce the *instantaneous functional requirement* or the objective of the design. The author finds this appropriate for the application area of mechanical transmission design. (It is also similar to the method used to describe the characteristics of electronic circuitry.) In subsequent work Chakrabarti et al., (1992), functions of systems are described in terms of a set of *generally valid functions*. Chakrabarti et al. see that the use of an input-output representation, as per Pahl & Beitz (1984) is restrictive, so they prefer to describe a function by the influence it exerts on a set of characteristics; type or outward form, magnitude or component of the type, number, place and time. This can be regarded as a type of state transformation. Other examples of systems represented in terms of inputs and outputs include Freeman & Newall (1971)and Rodenacker (1976).

By contrast, Ulrich and Seering (1989) use function objects to represent a manipulation of energy or matter between an input and an output representation in the form of icons. For each function object, the manipulation of energy and/or matter and has a flow and resistance to flow factor attributed to it. This indicates the manner in which the function object (referred to as a synthesis schema) manipulates the substance it carries. This is similar to the representation in Roth (1982) using a catalogue of *general functions* including *change, vary, connect, channel* and *store* for energy, materials and information.

A limitation of Ulrich and Seering's approach and others using this type of representation is that they suit systems designed to manipulate either materials, energy or information. However, for fixed structures and mechanisms it becomes difficult to represent function adequately. As an example, the function of a screw and bolt supporting a shelf or a box to protect its contents cannot be described easily with an input-output model.

This type of limitation firstly highlights the semantic gap between function and behaviour for some classes of objects, and starts to show a relationship between adequate representation and application.

4.3 Qualitative physics representation

The major applications of qualitative physics (QP) are concerned with the analysis of already constructed systems. Although there are few examples of its use for the synthesis of designs, many researchers acknowledge that the representation does have potential for application to design. Examples of work focusing on the issue of QP in engineering design include Forbus (1988a), de Kleer (1984 and de Kleer & Brown (1984), Williams (1984, 1991), Joskowicz 1988a,b) and Joskowicz & Addarbi (1988). Also, Forbus presents an excellent review of QP which discusses its application to design (Forbus, 1988b), and for general reading Bobrow (1985) and Weld & de Kleer (1990).

Work by de Kleer and Brown shows how components can be combined for configurational design. Forbus also presents the following explanation of their work, "A system is constructed from a collection of *devices*, like transistors and resistors. The behaviour of a device is specified by internal laws, often decomposed into distinct states or operating regions. Each device has some number of *ports*, and all integration between devices occurs through these ports. To model a specific system, one builds a network of devices. The device network is then analyzed by using the combined equations from the devices and interconnections". Within this work function is regarded as "what the device is used for". In support of this behaviour is "what the device does" and structure is "what the device is".

Forbus sees that this representation has the advantages that (i) it is efficient for computation (ii) given a correct set of models and connections an arbitrarily large system can be constructed, and (iii) it can be transferred across domains. However, there are also disadvantages; the approach offers no advice for the construction of models and also it is restricted to systems that deal with flow of some type.

An approach called *interaction based invention* by Williams (1991) illustrates the use qualitative descriptions about component based interactions. The components can be used to create devices that will achieve the required behaviours. The methodology has three major stages (shown in Figure 2(a)): (i) building the interaction topology, (ii) building the corresponding physical structure, and (iii) verifying that the desired interactions are produced. William's Ibis system initially develops a space of all the interactions for a given domain and technology and a space of all

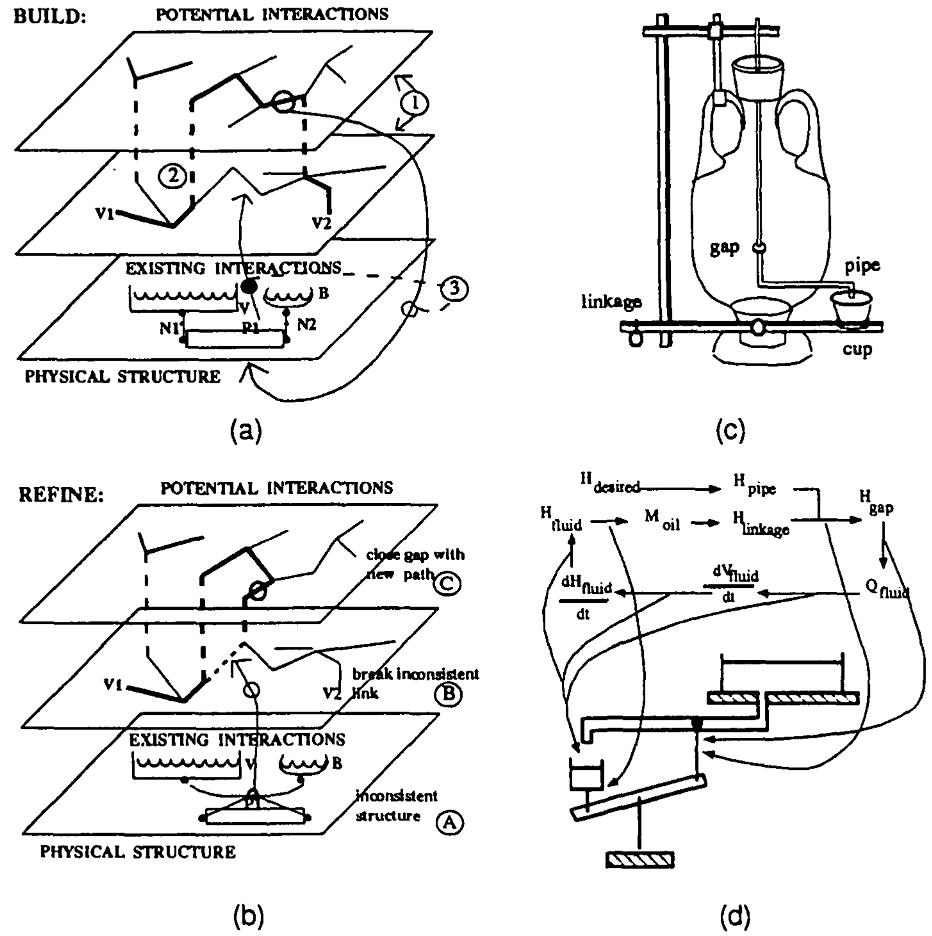


Figure 2 Interaction based qualitative design (Williams, 1991)

the existing interactions for a given structure. Next Ibis suggests a path which may connect up interactions from either space. Finally the path is verified to ensure it does produce the required behaviours.

In Figure 2(b) the refinement process is shown where a pre-defined path can be altered or an alternative path sought. In this case, Ibis identifies the inconsistencies, breaks the relevant links and finally closes the gaps to complete the chain. A graph can be produced of the interactions for a given system. Figures 2(c) and (d) show Heron's weight regulator and the corresponding interaction invention (plus interaction graph) that Ibis has created.

There are two well recognized alterative paradigms to device centred QP modelling. The first is the use of processes as primitives (Forbus, 1984, 1988c). In this case, physical changes could be described in terms of these processes, e.g. supply, transitions or consumption. Again, this representation, like the component based, is able to model system behaviours but does not explicitly indicate the functionality of a system. The second is the modelling of system behaviour with constraints (Kuipers, 1984, 1986). In this case, a system is designated a set of parameters. It is the change of these parameters within the given constraints that describes the behaviour of the system.

In general, QP representations have not been favoured by CAD system builders who wish to utilize knowledge of functionality. One reason for this is that knowledge of the complete device is required before it is possible to reason about how it will function.

The major contribution of QP work for working with functionality in design is in its potential for representing and reasoning with the mode of action of a device. Thus it is concerned with working

Table 1 Effort and flow variables for each energy domain (Gawthrop & Smith, 1992)

Domain	Effort	e	Flow		Momentum		Displacement	
Electric	e.m.f. (voltage)	e V	current	i	lines	λ	charge	q
Magnetic	m.m.f.	M	flux rate	$\dot{\phi}$	—	—	flux	ϕ
Hydraulic	pressure	P	volume flow rate	$\dot{V}$	pressure momentum	p	volume	V
Mechanics (trans)	force	F	velocity	V	momentum	P	displacement	x
Mechanics (rotation)	torque	T	angular velocity	ω	angular momentum		angle	α
Thermo-dynamics	temperature	T	entropy flow rate	$\dot{S}$	—	—	entropy	S

rather than purpose. The major challenge which faces researchers is to apply the formal methods to a process of design synthesis.

The influence of the QP work can be seen in a number approaches. A principle borrowed from QP is *Teleological Analysis*, the doctrine of ultimate purpose. De Kleer (1979) was the first to refer to teleology with respect to QP. He states:

> "Systems whose behaviour is to achieve a sole purpose—they have a *teleology*. By knowing the purpose of the device the interpretation whose behaviour is consistent with this purpose can be selected as the correct one."

Both Umeda et al. (1990) and Gero et al. (1991) suggest that the use of teleological analysis would help the designer make the subjective transition from purpose to behaviours. For example de Kleer's study of a trigger circuit identifies that there are four candidate outputs that could arise from such a system. However only two are feasible when the prestated operating conditions are applied.

4.4 Function represented with Bond Graphs

Bond Graphs were developed by Paynter (1961) and have since been well investigated by Karnopp and Rosenberg (1975). They provide a unified representation of physical systems spanning a range of applications by graphically depicting a system description in terms of an electrical analogue. This provides the representation with a well founded, formal basis which can be utilized across domains, as presented in Gawthrop & Smith (1990) (see Table 1). Bond graphs include four primitives, each a type of "energy manipulation" including sources of effort or flow, two types of storage and dissipators. Linking nodes are arrows indicating flow and causality. The movement of energy through the system is characterized in terms of *effort* and *flow*. In other work this is referred to as *per* and *trans*. A more detailed explanation of bond graphs is beyond the scope of this review. However, for an extensive account of bond graph theory and their usage refer to Karnopp & Smith (1990) and Gawthrop & Smith (1992).

Rinderle et al. (1989) use bond graphs to model electro-mechanical systems, including DC motors. The authors' select this representation because it allows the combination of modules into larger scale modules through the modelling of individual components as a module of bond graph, thus supporting modelling at different levels of abstraction. Although the bond graph representation does not explicitly represent the functionality of a system, it does indicate functionality implicitly by the integration of component level behaviour into device functionality.

In a discussion on the use of Bond Graphs, Finger and Rinderle (1989b) develop a representation of functionality in the mechanical domain. They claim their method provides information about how to make transformations from functionality to physical device descriptions. In support of using bond graphs they state the close relation to the structure of the system yet with enough information to generate alternative descriptions using other representations. In support of the use of energy representations is Cassirer's (1923) argument "energy provides a unifying measure across physical domains".

Gawthrop et al. (Gawthrop & Smith, 1990, 1992; Gawthrop, 1989; Gawthrop et al., 1990) advocates the application of bond-graphs to systems control including both continuous and discrete events. Control is complex in nature working at a number of levels of abstraction with unclear specifications of the impact of localized events upon global events. Furthermore, the technology and specification change with time. The need they perceive in this area is a control system to specify required performance for the system. Evidence suggests that there is a need for an environment to design control systems. Systems may include gas turbines, distillation columns and company economics.

Gawthrop concisely summarizes the benefits of this representation and the advantages it has over input-output block diagrams:

- there is a correspondence between the bond graph and the physical system it models,
- graphs can be created before the causality is considered,
- sign conversions can be adopted,
- at a high level it provides a symbolic representation,
- it may provide a symbolic or declarative approach to modelling,
- graphs make a clear distinction between behaviour and structure,
- graphs can be constructed independent of system state.

A major conclusion from the work on bond graphs is that it is seen as a formalism which supports the representation of function as mode of action, but that it can be applied to a wider variety of artifact descriptions than input-output representations. However, the formalizing of problems as bond graphs can present problems.

4.5 Representation summary

In conclusion to the representation discussion, a chart (see Table 2) presents the main features, advantages and disadvantages of the representations that comprise the focus of this section, namely; function grammars, input-output, qualitative physics and Bond Graphs.

5 Manipulation of information about functionality

To be of value within a computer based system, any representation has to have corresponding reasoning mechanisms. The major concern is this section, however, is to highlight the types of reasoning approaches which are relevant to overall manipulation of functionality. Within this section different reasoning methods are examined and the circumstances under which each method is used are identified. Examples are given of the ways in which various authors execute the types of knowledge manipulation. For the purpose of the review three types of knowledge manipulation are considered.

1. *Combination*—the union of one function with another function.
2. *Breakdown*—the "division" of a function into a number of functions that achieve the same action or purpose.
3. *Classification*—the grouping of functions into groups on the basis that they can be used to serve the same actions or purposes.

These together provide a basic toolkit for function manipulation.

5.1 Combining functions

"The combination or integration of functions can be regarded as good design", is the proposition put forward by Finger and Rinderle (1989b) in their work with bond graphs. It is their aim to support designers by developing a method for the integration of complete or partial previous design cases to currently evolving design cases. They recognize the following, "Transformations that

Table 2. Comparison chart for principal representations.

Representation	Principal Concepts	Advantages	Disadvantages
Input/output	*Flow of energy, matter and information into and out of systems. The system is characterized in terms of the state of the output relative to state of the output	*I/O relations useful for defining primitive types of energy manipulation *can build composite systems from smaller ones *has implied causality *facilitates abstraction	*limited to systems incorporating some aspect of flow *decompostion tends to be subjective
Gramatical representations	*The definition of function using, for example a verb–noun pair label e.g. 'transfer-torque'	*can be built into a systematic library *intuitive *transferrable/terminology independent, may be used across domains	*can restrict creativity *ambiguous—can describe either purpose or action *provides no information to support any kind of representation manipulation
Qualitative physics	*Symbols (or sets of), describing operation of a system in terms of processes, components or constraints in order to describe/predict behaviours over discrete intervals of values	*can describe systems that cannot be quantified *domain independent *correspondence to equivalent physical system	*cannot model large systems *suffers from discerning between possible and not possible behaviours *generates large number of possible outcomes *requires a lot of information about a complete system
Bond graphs	*Electrically analogue the system's operation. Includes a primitive set of energy manipulators (nodes). The nodes are linked by bonds each describing the character of energy passed between nodes	*allows *post priori* causality allocation *symbolic representation *enables distinction between behaviour and structure *can build composite systems from smaller ones using rigorous physical principles	*limited to systems incorporating some aspect of flow *no support for abstraction *no explicit indication of function

result in increased functional integration create a device configuration in which a single component contributes to more than one of the behavioural requirements of the design." To achieve this they use predefined transformation relationships or graph productions. These are stated in the form:

$$P = (g_L, g_R, E)$$

This relation represents the left (g_L) and right g_R) sides of a production and an embedding transformation (E). It provides a replacement operation in the context of graphs. After the

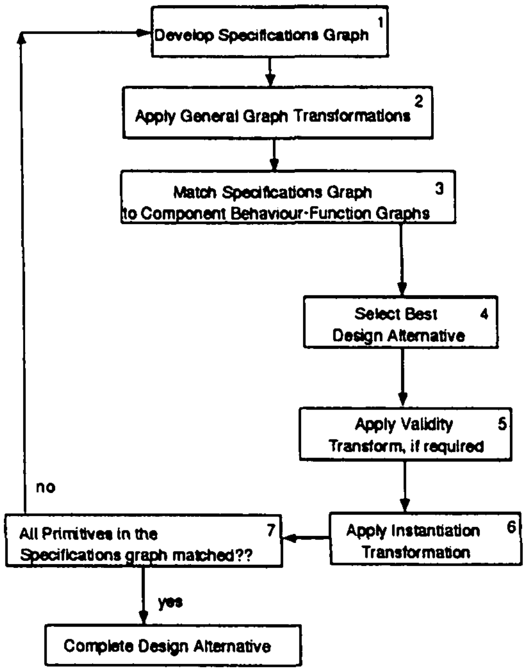


Figure 3 Synthesis process for bond graph embedding transformations (Hoover & Rinderle, 1989)

removal of the left part of bond graph g_L from its containing graph, the right hand graph g_R (isomorphically equivalent to the left) can be inserted. They use this transformational strategy to make changes to evolving design solutions. Thus using the embedding transformation they can make transformations from graphs containing the graph fragment g_L to one containing the fragment g_R. This basic operator allows construction of a functional reasoning engine, which is goal driven or generate and test driven. The transformation process is described by Hoover & Rinderle (1989) (Figure 3).

The approach taken in Freeman and Newell (1971) is based on a set of propositions. Examples of those related to function combination are as quoted (*sic*) (see also section 6):

- A functional connection can occur between two structures if one provides a function required by the other (This proposition is similar to Elm's notion of functional support in the case of a supporting function providing a level of performance to another function.)
- A constructed structure consists of a set of structures (its parts) and a set of functional connections between them such that:
  (a) the functions provided are those provided by the parts that are not consumed in functional connections,
  (b) the functions required are those required by the parts that are not provided by a functional connection.

The above proposition means some functions may be internal to a system and therefore may not be offered to the environment by the system (i.e. they disappear "into a black box"). Also the required functions are those not already supplied by another function. Their example of a knife (Figure 4) is as follows;

> A functional connection exists between the blade's requirement for being held (R1) and the handle's provision of that function (P2). The blades provision of the cutting function (P1) is not consumed in a connection and is thus provided by the constructed structure; likewise the handle's requirement of being held (R2) is not satisfied and is thus a functional requirement of the knife."

Further to the above propositions, they also acknowledge a constructed structure must obey the restrictions on its subparts after the effects of the functional connections are accounted for.

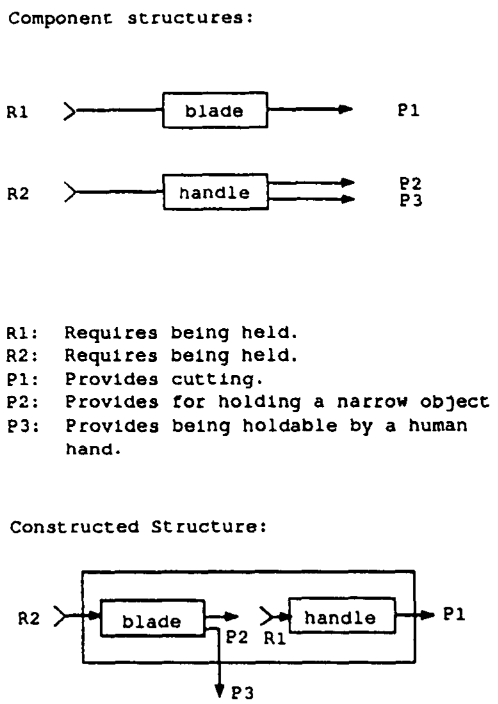


Figure 4 Construction of functional structures (Freeman & Newell, 1971)

In his Function Structure Methodology, Elm (1988) sees functions can be combined through the *support* relationship. This relationship recognizes that a supported function requires some level of performance from a supporting function. This support may be either *minimal performance* or a *change of performance*. In Taura & Yoshikawa (1991), the operation of moving from the *partial function space* to *total function space* is the equivalent of combining functions. In terms of design operations, they see this as the combination of functions each of which is representative of a view of the design model (see section 5.2). Chakrabarti et al. (1992) regard "support" as central to the combination of functions. To achieve this they use functions of support for a behaviour. That is, to provide a specified behaviour a specified function, or set of functions, are required.

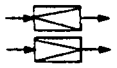
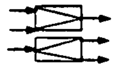
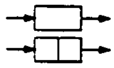
An explanation of representation of *multiple functions* is provided by Eastman et al. (1989) because, as they state, in almost all cases a product is required to provide more than one function. This is deemed necessary to support subsequent design analysis. Eastman et al. achieve functional combination through the combination of functional hierarchies (see section 5.2).

5.2 *Function breakdown*

The most common use of knowledge about functionality in support of the reasoning process in engineering design is for functional decomposition. Most workers hold a viewpoint similar to Roth and Rodenacker (see section 3); that the design process is initiated by a functional requirement. Furthermore design proceeds by the development of a function structure. This is through the decomposition of a functional requirement into sub-functions. Decomposition of function(s) continues until there is a correspondence between the functions and components. In Roth's case, the aim is to subdivide functions until there is more or less a one-to-one mapping. Eastman et al.(1989) also use function decomposition to support component-function mapping. They see it as supporting functional analysis of a design. However, they offer no mechanism for this operation because of its conceptual and abstract nature.

Taura and Yoshikawa (1991) regard mapping from *total function space* to *partial function space* as *functional decomposition*. The nature of the decomposition varies, because to go from *Total*

Table 3. Generally valid functions (Chakrabarti et al., 1992; Pahl & Beitz, 1984).

Characteristic	Generally valid functions	Symbols	Explanations Inputs (I)/Outputs (O)
Type	Change		Type and outward form of I and O differ
Magnitude	Vary		I < O O > I
Number	Connect		Number of I > O Number of I < O
Place	Channel		Place of I ≠ O Place of I = O
Time	Store		Time of I ≠ O

function space to *Partial function space* the designer requires to take a *view* of the evolving design solution. Umeda also agrees "in principle" to this strategy.

Subsequent discussion presents a mathematical model of the identity of function space. This discussion shows the relationship between the members of any sets that comprise the ideal function concept set. It also provides definition of the total function concept and the total function space. In their model of functional operation, they see function decomposition as the mapping of design entity space in partial function space onto an entity concept in total function space.

5.3 Classifying functions

"In experience based design, new problems are solved by retrieving and adapting the design cases encountered in the past." This is the view point taken by Goel (1991). The work describes a scheme for organizing previous design cases according to the functionality they deliver. Alternatively, Eastman et al.,1989, Eastman 1982,1987 uses *generalization* as an important form of data abstraction. He treats generalization in two distinct forms *combination* and *classification*. The former being a form of grouping for both similar and dissimilar functions. In contrast, classification is achieved through the grouping of individual instances according to predefined common attributes.

In their work, Chakrabarti et al. have chosen to represent functionality as a change in a set of characteristics representing the input and output of the function (and not in terms of the input-output itself). The characteristics; type or outward form, magnitude or component of the type, number place and time can all be changed, as shown in Table 3. The identified *generally valid functions* provides a genetic classification into which all functions can be placed.

As part of a strategy to support design adaptation, Goel & Chandrasekaran (1989) classify functions. They do this through the matching of behaviours that are regarded as the output of their representational schema. The matching of the outputs to inputs, in this case between schema, is one of the more common methods of function combination. In this case, as with others, it is regarded as providing an indication of causal directionality (see de Kleer & Brown 1980 or Iwasaki and Simon 1952, 1986a,b. Others workers using the input/output representation utilize a similar method to that of Goel et al. include Rasmussen (1984), Rodenacker (1976), Roth (1982) and Ulrich & Seering (1989).

5.4 Manipulation summary

This section has addressed the variety of approaches taken to reasoning with functional structures within a design context. These have been organized into function combination, function break-down and function classification. In practice all three categories are complementary, but also particularly appropriate to function interpreted as external behaviour, and in situations where it is possible to identify a hierarchical breakdown of functions through the inherent organizational or logical structure of a design. It is not always possible to determine such structures in advance of developing a design solution unless the artifact belongs to a well defined class of objects which allow re-use of past structural information.

6 Design framework relations

A framework in the context of this paper is a set of relationships between function and other design characteristics; in particular structure and behaviour. Emphasis is placed on identifying the characteristics which authors use for describing and developing a model of a design. It is with these frameworks that CAD systems will be organized to allow designers to "move about" during the design process. As the designer wishes to develop a design he/she may do so by altering any characteristic or set of characteristics. Using the framework, the designer can then determine how the other aspects of a design will change; in accordance to the pre-determined relationships. Furthermore, the way a framework is constructed can constrain the evolution of a design.

A clear expression of the relationship existing between functionality and other design characteristics is given by Simon (1982):

> "The fulfilment of purpose involves a relation among three terms: purpose or goal, the character of the artifact and the environment in which the artifact performs."

The statement also gives an indication of dependence; that for the function (in Simon's case purpose) to be achieved then the designer must appreciate the nature of the other factors involved.

Simon defines a model in terms of the relationships between an artifact, its purpose and its outer environment. If the inner environment is appropriate to the outer (or *vice versa*) the artifact will serve its intended purpose. Predictions can sometimes be made without detailed assumptions of inner environment. There are some advantages to using an approach like this to describe systems. The first advantage of dividing outer from inner environment is that we can predict possible behaviours which may occur. Another advantage is that the factors influencing the success of an artifact in achieving its purpose usually rests with a few characteristics of the outer environment and not the inner. Also of note is that we often find different inner environments accomplish similar goals.

An example centred on the function—process relationship comes from Elm (1988). He sees functionality as a collection of processes. The framework has a function consisting of inflow, conversion and outflow processes. It is the flow of a commodity or commodities through these processes that enables a process to deliver the stipulated function. Associated with each process is a set of process elements. The types of process elements vary depending upon the type of process (see figure 5). It is these process elements that relate quite closely to components of an equivalent system and it is this basis that forms the informal relationship between process and component. A given structure can be related to other function structures through a *supports* link. The link enables the development of large scale systems to be described. It relates a function at the process element level to another function. The link provides information about the way in which a given function must supply a process element. For example, a process element supplies high pressure steam. Through the use of support links a constraint may be set up to show at what rate the supporting function should supply the steam.

Eastman et al. (1989) developed the Engineering Data Model (EDM) for defining schemas for integrated engineering databases. The models represent knowledge of function and relate it to knowledge about a design's form and physical properties (Figure 6). They are designed to support multiple level abstraction and represent semantics that determine the validity of the information. Also a set of class structures to support functional descriptions, definition of products, generalizations of objects and functions, constraint structures and structures able to depict engineering technologies.

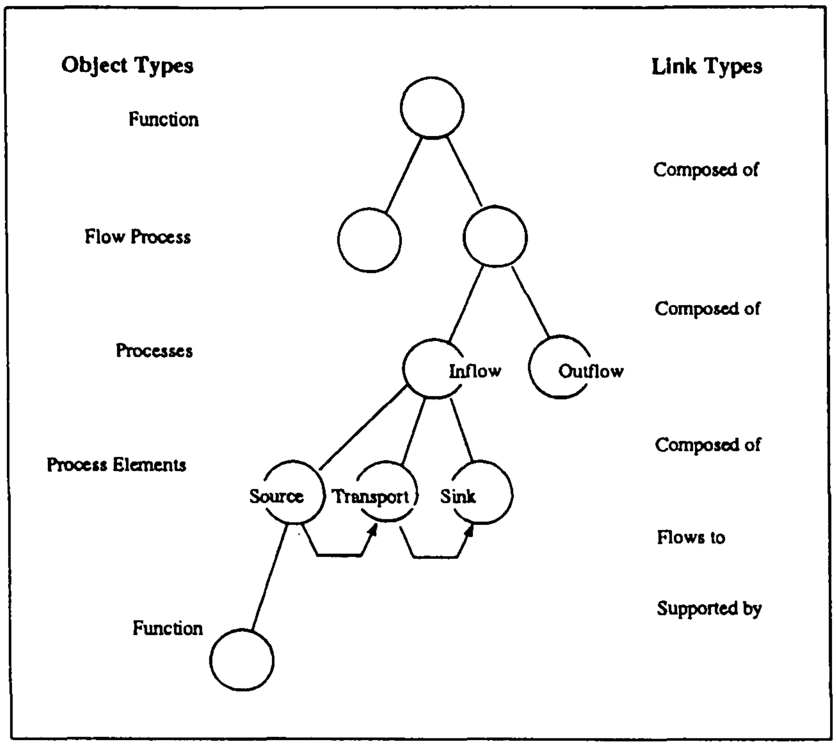


Figure 5 Functional decomposition—Developed by Westinghouse Control Systems (Elm, 1988)

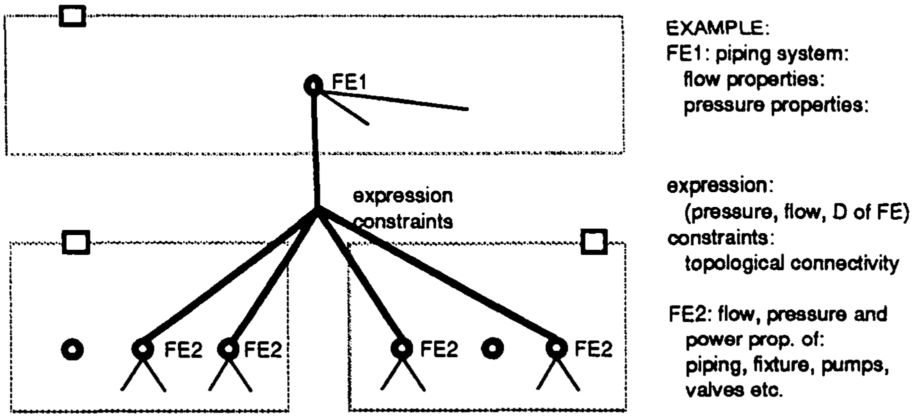


Figure 6 Accumulating functions—Relating FE's from several PO's (Eastman et al., 1989)

A conceptual basis is provided by Eastman as a precursor to explaining the relationships in his data-model. Here he discusses *functional analysis* stating it is the functionality of a product that determines whether its properties are relevant, and that functionality in this work is considered from the widest possible perspective and are necessarily defined at multiple levels of aggregation. Next, *Physical Objects* are considered; their inclusion often imposes strong constraints on the manipulation of function information. *Generalizations* from Smith & Smith (1977) enable the model to be decomposed into components. Also used within the model are *constraints* dictating the range of values a product's variables may hold. Finally, the principal structures themselves are *data models*, *classes* and *instances*.

From the above concepts, Eastman et al. define the set of three structures comprising the product information structures, namely;

- *Functional Entity (FE)*—an hierarchical aggregation of functions with pertinent constraints.

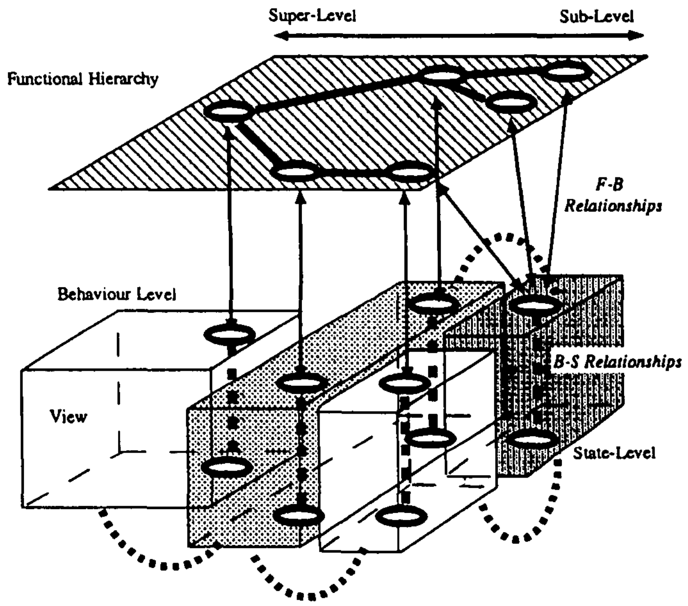


Figure 7 Function, behaviour, structure diagram (Umeda et al., 1990)

- *Physical Objects (PO)*—an hierarchical aggregation including sets of FEs and constraints are bound to the PO.
- *Generalization*—an hierarchical aggregation of POs that acts as a form of data abstraction. Note that specialization is used as the inverse form.

The FE is included within the PO and a PO may include multiple FEs that are different views of similar data. To combine the above structures two relationships are used, *instance-of* and *iaggregation*, the latter being the aggregation of instances of a structure. These structures are then linked into the *engineering data model* (EDM). In doing so, there is a set of criteria to which an engineering model must respond. Those relevant to this work include:

- *Representation of function*—by function the author means the purposes which the product addresses and the means to evaluate them. He sees the relation of form to function as critical to recording the basis for design decisions. More generally the representation of *design intent* is of interest.
- *Support for multiple levels of abstraction*—the need for abstraction is stated. Two issues here are important, the maintenance and extension of the set of views.
- *Open-ended semantics*—EDMs should support extension and change of their underlying conceptual structure. These semantics overcome the limitations of standard database structures.

Umeda et al. (1990) have created a more formal definition of a framework as a *Function Behaviour Structure* (FBS) relation (Figure 7). Within this framework they have defined function, structure and behaviour along with their supporting relationships. The framework facilitates the transition from a statement of function to a statement of form by using knowledge of design behaviours.

They model the physical world using the concept of the internal state S_i and set of states S represented in terms of entities E, attributes A and relations R:

$$S = < E, A, R>$$

$$S = < S_i, S_o$$

The transition from one state S_1 to another S_2 is the basis of their definition of behaviour—"sequential one or more changes of state." This is governed by physical laws, a physical law being

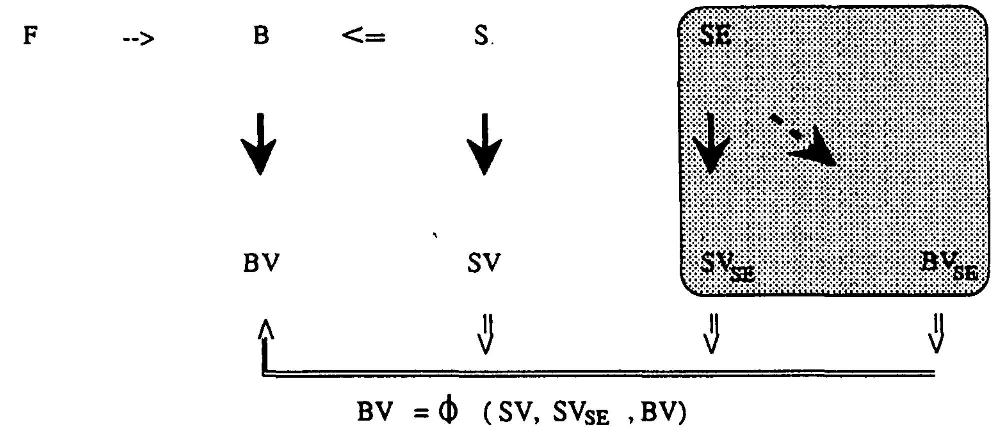


Figure 8 Relational framework for function, behaviour and structure (Gero et al., 1991)

"a rule which determines behaviours of an entity under specific conditions of states." From this they can generate all possible behaviours from physical laws. This is their *B-S Relationship*.

To account for the influence of physical situations on the behaviour of a system they have introduced *views*—a specific representation scheme for behaviours and states. From their notion that function is an "image of behaviours abstracted by humans", they set up their *F-B* or "function-behaviour" relationship. Like de Kleer and Gero, they too suggest the use of the teleological analysis method as used in device centred qualitative physics (see section 4.3) to make the transition denoted by Γ_{ab}:

$$\Gamma_{ab}: B \rightarrow F$$

Gero et al. have a variation on the function/structure/behaviour relations with the following definition:

- *function*—the design intentions or purposes,
- *behaviour*—how the structure of an artifact achieves its functions,
- *structure*—the components which make up an artifact and their relationships.

Based on the assumption that most CAD systems only deal with structure, or at best have function to structure coupling, they believe this is not a sufficient approach. They argue that in design, only behaviours with function-related performance are considered and knowing the behaviours of structures makes the choice of function less intuitive.

Their framework relates function, behaviour and structure for which they state function seldom provides an evaluable basis for analysis of performance. Instead, the use of teleology is advocated.

Behaviours are characterized by a set of behaviour variables, acting as descriptors of the performance, becoming relevant when the behaviours are a means to the required functions. Likewise, structure is characterized by structural elements (materials, geometry, etc.). To configure components they believe structure variables should be configured. The creation of variables is simply parametrization. To link behaviour to structure, use structure, structure elements and behaviour variables.

In their work, Finger and Rinderle (1989b) see interactions of components as important to a synthesis strategy, and argue therefore that there should be a link between behaviours and physical characteristics. To achieve this end they transform an abstract description of behaviours into a specification of physical components. For this they chose to represent specifications, components and devices.

They identify a set of requirements for their approach stating that it should:

- express formally the design requirements,
- be compatible with the representation of the behaviour of components,

- facilitate the validation of the design,
- represent the required system behaviour without imposing a pre-defined structure on the physical realization of the design.

The starting point in a design case is a specification including behavioural and physical information. These characteristics are regarded as independent in the functional domain, but are related in the physical domain. Their argument is that for any given component set there will always be a set of behaviours. The authors also note there is a dependence between the physical and behavioural characteristics of any given component. The components are represented by combining abstractions of their primitives. Note, individually these may not map onto components or sets of components. That is, Finger and Rinderle believe that one-to-one mappings are rarely appropriate. However, collections may be related to a component or components; in essence a many-to-many mapping which in practice is often the case. The transformations associated with these relationships include "behaviour preserving" and "component directed". These relationships aid selection and configuration of components based on:

- *functional integration*—the integration of functionality in order to satisfy more than one behavioural requirement,
- *utilization of incidental behaviours*—when incidental behaviours are utilized leading to the exploitation of secondary behaviours or component characteristics.

A succinct description of relational structure is provided Freeman and Newell (1971) (see also section 5). They use a set of propositions to describe a 'model of the design task environment' thus;

P1 Each structure provides a set of functions.
P2 For each function it provides, a structure requires a set of functions.
P3 A functional connection can occur between two structures if one provides a function required by the other.
P4 A constructed structure consists of a set of structures (its parts) and a set of functional connections between them such that:
  (1) The functions provided are those provided by parts that are not consumed in functional connections.
  (2) The functions required are those required by the parts that are not provided by a functional connection.

In contrast to other work, the authors directly manipulate knowledge of functionality. Other systems rely on the mappings and translations between knowledge of functionality and other characteristics because they see that reasoning about functionality alone is not feasible. The aim here is to support that stage of the design process when the designer will think in terms of function either predominantly or totally. The principal linking relationship between functions, the provides/requires relation, may have various types rules or laws applied across it in a similar manner to the support links in use in Elm (1988).

Rasmussen (1984) summarizes the features of an abstraction based hierarchy for representing and manipulating the functionality of process based systems (Figure 9). Note he views these systems as dynamic, and thus the perception of the decision maker when reasoning about the system changes constantly. On this basis, Rasmussen suggests a structural representation of a physical system is a suitable framework for the FSM. Within the framework an object structure is represented using "part-whole" relations, and process level cause/effect relations can be used to predict the course of events through a system.

When modelling physical systems he observes that their descriptions can vary between abstract and concrete by means-ends relation for relational concepts. In this abstract-concrete dimension properties can be represented over several layers of abstraction, the lowest level being physical

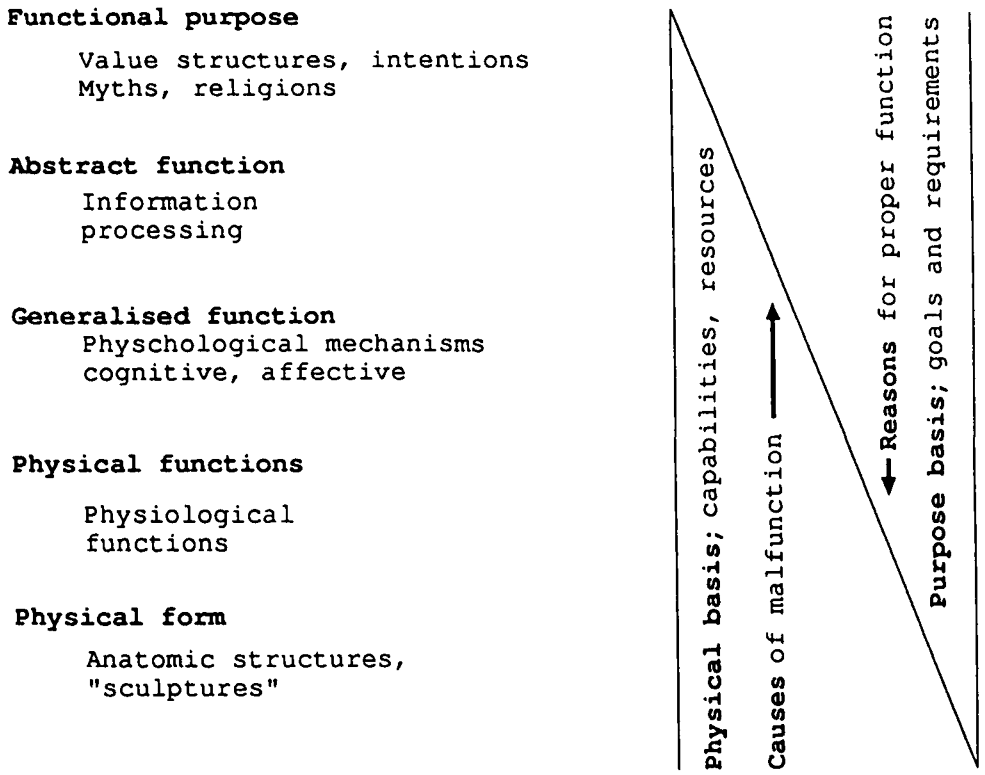


Figure 9 Means-ends abstraction hierarchy for representation of functional properties (Rasmussen, 1984)

component detail, and ranging up to functional aspects of purpose at the highest levels. To represent levels of detail part-whole relations are used. The resolution of the description is controlled by the aggregation and decomposition of elements in the levels above and below.

Within the model there are five distinct levels of abstraction. From the least abstract to the most abstract they are;

- *Physical form*—the most concrete level contains information on physical appearance and configuration of systems and its parts helps identification of the system components.
- *Physical function*—process based information relating to physical implementation, described as physical components and variables used to characterize their functional states, e.g. a diesel engine and its combustion cycle.
- *Generalized function*—abstraction from representation of function of a discrete component, function is represented at a composite level, e.g. a frequency amplifier is represented by its gain and selectivity curves and not as a group of transistors and oscillators. Generalized functions are implementation-independent and tend to be abstractions of these specific implementations. Trying to establish relationships between this level and the physical form level is difficult. There is rarely a one-to-one relationship between the two.
- *Abstract function*—includes information about flow of energy, mass, intended operating state, conditions of operation (for example laws of conservation) and causality.
- *System purpose*—information about intended system functionality in a specific environment. Quantitative input-output expressions can be sufficient for this description.

As an explanation of the above, Rasmussen makes a set of observations relating to the use of abstraction hierarchies. He believes abstraction is not simply a case of removing information of physical properties. Information is added according to the higher level principles governing the co-functioning of the lower level elements of the hierarchy. These higher level principles are influenced by the system's purpose. This is in line with the concepts underlying teleological analysis.

Rasmussen also comments upon the character of the models themselves. He notes models constructed at a high level of abstraction have general applicability and can serve several purposes. Alternatively, models at high levels of abstraction are more specific in their applications. In this context, a high level model may represent the functionality of a system whereas a lower level model may only represent a single operating state of the same system. Furthermore, the information influencing the 'proper' functionality of a process is from the higher levels of abstraction, whereas information about process limitations is found at lower levels of abstraction.

Fundamental to Taura and Yoshikawa's (1991) work are the following definitions:

- *entity concept*—an entity with attributes,
- *attributes*—being properties (for example physical, chemical, etc.),
- *behaviour*—is regarded as a change in attribute value,
- *function concepts*—are derivatives of attributes or behaviour from the application of rules.

There are also *total function concepts*—a derivative of behaviour and in contrast are *partial function concepts*—derived from components. They regard the design process as a cycle through a series of tasks from a specification, followed by data-retrieval, function analysis and then verification. This terminates on creating a design functionally equivalent to the specification.

It is clear from the various approaches covered in this section that the design context, and in particular, consideration of the design object has a major influence on the handling of function and functionality. The interaction between function as design purpose, structure as a structured representation of the design, and behaviour as a description of the external performance of a design underpins all the attempts to support function within CAD systems. Approaches vary, sometimes in the emphasis on structure or behaviour, instead of function. Other differences are more philosophical and stem from different definitions of function; generally, whether function can be mapped directly to external behaviour or exists as some conceptual idea which can only be evaluated by extrapolating knowledge of behaviour in an environmental context.

The next section will develop these ideas and distinctions further through consideration of the design process.

7 Models of the design process

Previous discussion of perception, representation and manipulation of functionality gives an indication of areas of variation and consensus between the practices of design engineers and CAD system developers. An overview of relational frameworks shows how this knowledge of functionality can be related to other characteristics of an engineering design. To make use of these findings within a CAD system it is necessary to see how they relate to the design process. Furthermore, studying models of the design process gives insight into the following:

1. Methods for translating functional requirements into structural specifications.
2. The stages of the design process when knowledge of a design's functionality makes a contribution to the design process.

A widely accepted model of the design process is found in the work of Rodenacker (1976) (Figure 10). Although developed for process engineering, it can be adapted to other areas. The underlying assumption is that every design must fulfil required purposes or functions and design is the transformation of information taking the designer from the abstract to the concrete; from function to form. Many other design models represent a variation of Rodenacker's work, as shown in this section:

1. The initial task is to establish a function structure, an abstraction of the design's requirements.
2. Next is the selection of an appropriate physical process.
3. Finally, a set of required form features are found.

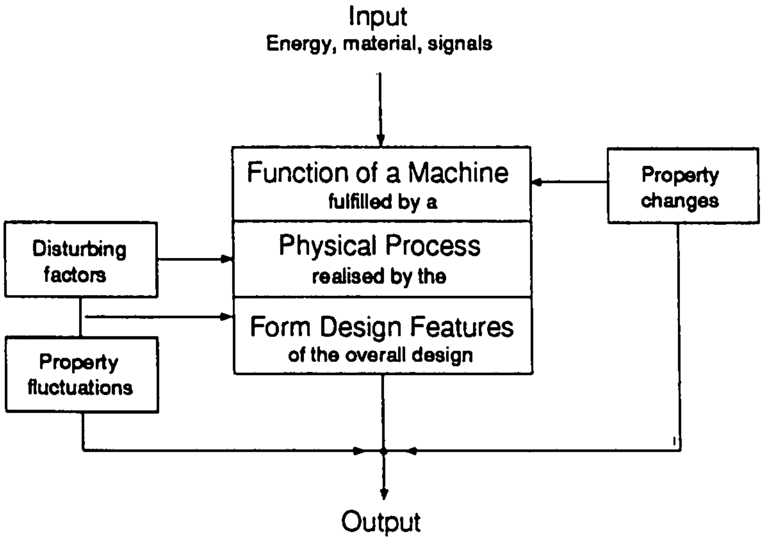


Figure 10 Design steps according to Rodenacker (Pahl & Beitz, 1984)

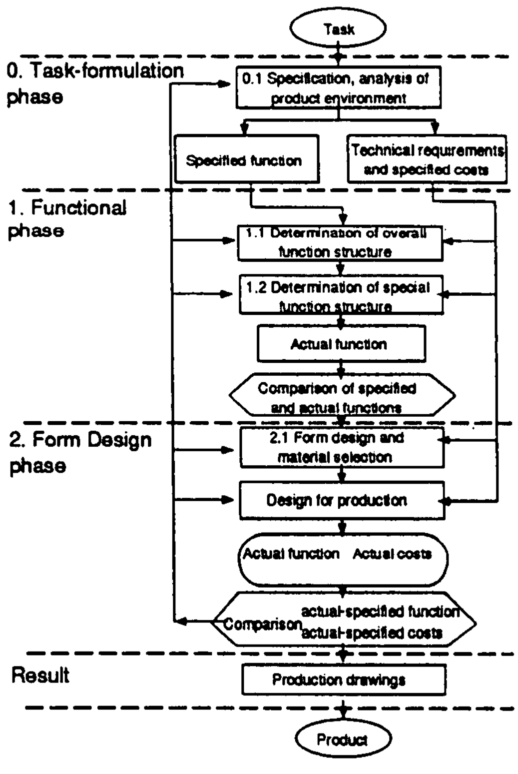


Figure 11 Phases and steps of the design process (Pahl & Beitz, 1984)

Supplementary to the above Rodenacker acknowledges the basic design process is subject to external influences including disturbing factors, property fluctuations and property changes.

The design process model developed by Roth (1982) requires iterations through each of the phases (Figure 11). This is in contrast to Rodenacker's, where iterations are carried out across the complete process. Initially, the problem is precisely evaluated leading to a problem definition including required functionality, technical specification and costs. Subsequently, appropriate functionality is selected from catalogues to supply each of the specification requirements (see section 4.2). After the nomination of the functions, combinations of sub-functions can then be sought, each representing a sub-problem in itself.

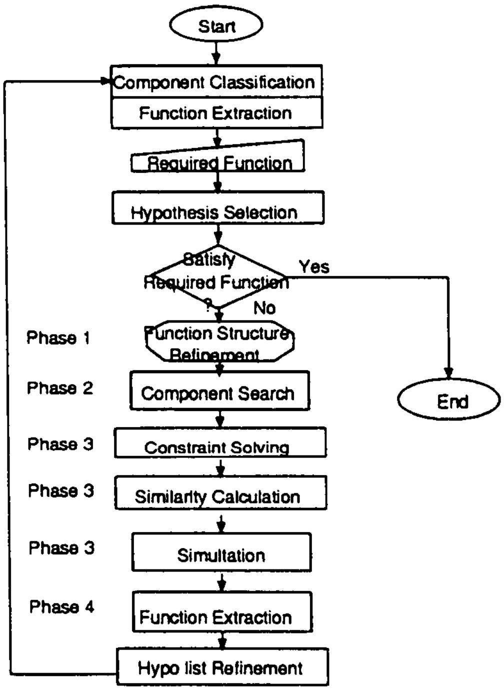


Figure 12 Algorithm for the generation of design solutions (Taura & Yoshikawa, 1991)

In their model of the design process, Taura and Yoshikawa (1991) have identified four principal steps (Figure 12):

1. *Function decomposition*—to make a complete set of function requirements (total function space) and select a sub-set according to a single viewpoint (partial function space).
2. *Component search*—the selection of components from a set of components (a component space), a library (perhaps) able to exhibit the functionality expressed in the partial function space.
3. *Component composition*—the construction of a rough component solution (an attribute space) from a component space.
4. *Evaluation*—comparison of the functionality that the component of the attribute space can exhibit with partial function space.

Subsequent to the four stage design process model, the authors put forward an algorithm for generating a design solution based upon the above tasks. An important feature of this model is the recognition that design incorporates an aspect of search. Essential to this is that the designer starts by considering the design in terms of the overall functionality required. The designer later evaluates the design's detail in terms of a selection of the design's functionality from a given viewpoint (partial function space).

Ulrich & Seering's (1989) model of the design process is an attempt to synthesize a system from an established set of components. Using as its basis the connections of the individuals by matching

- Physical system
- $Transformation_1 \Rightarrow Representation_1$
- $Transformation_2 \Rightarrow Representation_2$
- . . .
- $Transformation_n \Rightarrow Model$

Figure 13 System modelling (Gawthrop & Smith, 1990)

inputs and outputs the authors view the design process as a three stage process: (i) the construction of a rough working solution; (ii) the analysis of the behaviours of the solution; and (iii) the revision of the solution. It is their stage (iii) that is significant for it is prior to this stage they translate the component information in their input-output representation to make the necessary adjustment to the design solution.

Gawthrop & Smith (1990) (Figure 13) regard system modelling as a sequence of transformations taking a representation through a series of evolutionary stages until a satisfactory system model is achieved. They view the first transformation as the most difficult, dependent upon the engineer. Therefore *Transformation*$_1$ should not be automated. However, he does advocate that the other transformations should be automated where possible. Further to this, *Representation*$_1$ should form a basis for any and all of the potential models that will be generated.

Freeman & Newell provide an explanation of a "design path" to take the designer from design goal to a complete design while using functional reasoning. The authors advocate the use of functional reasoning as in the "General Problem Solver" (GPS) system (Ernst and Newell 1969). GPS is based upon a number of different types of means-ends analysis. It expresses a situation in relation to a goal, and operators in relation to how they affect situations. Next the operator can offer a function to fulfil the requirement and finally the system creates another function situation and puts forward its requirements once again. Termination will be when a situation occurs that the system deems not to have functional requirements. To undertake this operation they have developed the XDA CAD system that utilizes a model of the design process with the following steps:

1. The 'structure' of the design is defined by stating the functions it is to provide.
2. The designer is then given a choice as to which part of the design he wants to work on.
3. A set of possible structures are presented to him.
4. He may stipulate a structure to supply the function.
5. Its functional requirements are entered into the design and the cycle repeats.

This proceeds until a primitive structure is stipulated to provide a needed function, terminating the design.

In essence, they argue that the most fundamental model of the design process starts with a given set of structures and their functional specifications which leads to the construction of a structure with the desired specifications. This is also similar to Roth's ideas. Top-down or bottom-up schemes can be used, utilizing combinatorial/heuristic search. Top down alternates between selection of structures to supply functions and generating new functional requirements from the chosen structures.

Gero et al. (1991) also see design as "the process of moving from function (intended purpose) to structure (the components that go to make up a design)". However, central to their model is the use of behaviour as an important link between function and structure. The paper presents a model for routine design. Further to this, they view design synthesis as the process of proposing design solutions in terms of structure to satisfy requirements by the exploration of a general model. They also see design analysis as the prediction of potential behaviours of synthesized designs, the predicted behaviour being computed from values of structure characteristics (structure variables).

They see that the relational framework they have developed as supporting the synthesis and analysis tasks in design (Figure 8). The first step of design interpretation is recognizing candidate components and the development of component models that explicitly describe the artifact's behaviour. Here they reason, that the performance of a design is evaluated by analysis of components and interconnections. Also qualitative physics and teleology provide a basis for the derivation of behaviour from structure. From QP they define a process model of design, using a notion of *Design Context*. Their first principle is that *specific behaviour* for the artifact is derived from *possible behaviour* by *teleological knowledge*.

Each artifact belongs to a class represented by a generalized model. This is a knowledge source for structural and behavioural knowledge. For any class of artifacts there is an implicit relation between function and behaviour, hence *function* can be derived from *desired behaviour* with teleological knowledge. This permits establishing component behaviour relations and identification of structure in response to function. Thus *desired behaviours* can be derived from *structure descriptions* and *component specific behaviours* with knowledge of the structure and component behaviour. Furthermore, *component specific behaviours* and structure may indicate *actual behaviours* through causal knowledge. To evaluate the design *desired behaviours* are compared to *actual behaviours*. This forms part of an evaluation loop which includes feedback from relations in terms of refinement, structural and component behaviour knowledge.

To aid the subjective part of the design process Umeda et al. (1990) suggest a method starting with the construction of a hierarchy based on selected views (see section 6). When using the FBS framework, either the hierarchy or the views can be selected first. Note, the construction of views is based on both the intentions and functional parts. The eventual feasibility of the hierarchy is determined by the physical laws. They look at the FBS framework as a modelling strategy, stating it is necessary to model functions because design proceeds in terms of functions which leads to the recognition of entities and viewpoints. With respect to design they see designers using the types of strategy that they advocate the FBS diagram supports, that is, the division of function and the definition of behaviours/states that can produce the functions. However, they do not advocate the use of Rodenacker's one-to-one mapping; instead they suggest the use of Ulrich's function sharing approach. They believe a designer should collect relationships between functions and not the functions themselves.

Goel (1991) takes as input a physical device that achieves a required functionality and a functional specification for a desired device. The goal is to produce a structure to deliver the functionality. Stored cases include knowledge of structure, and a pointer to a structure-behaviour-function model. Cases are indexed by the functionality they deliver. Design specific structure-behaviour-function models form the system's adaptation strategy along with modification plans. Structure is viewed as constituted of components and substances and the functioning is in terms of flow of substances among the components.

The system, called KRITIK, is designed to tackle a number of design tasks:

1. *Case retrieval*—from the input function requirement retrieve a case that provides the closest equivalent functionality. For this they use a function based indexing system. Initially cases making a match, even a partial one, are selected. If more than one are retrieved, KRITIK will select the closest one to the required function. Closeness is defined by the number of features, slots in a behaviour state showing a variation (for example location, substance, etc.). For functional specifications that differ by the same amount the KRITIK will use heuristics to select one.
2. *Design Adaptation*—Two types of adaptation knowledge are utilized, declarative knowledge represented as structure-function-behaviour models and procedural knowledge as modification plans. Plan selection is based on the differences between two functions, the desired and the delivered function—a type of means ends analysis.
3. *Modelling*—The component-structure model is constructed from components, structural relations and substances. The model provides an indication of behaviour, the components can show internal causal behaviours with the substances.

Rinderle et al.'s (1989) understanding of the design process is expressed as "During the design process the designer transforms an abstract functional description for a device into a physical description that satisfies the functional requirements." In this sense, design is a transformation from the functional domain to the physical domain. Experienced designers will shortcut detailed work by recognizing existing relationships governing performance and configuration. Rinderle makes use of this assumption in his work through the understanding that design occurs at varying levels of detail and a design environment should allow flexibility in resolution as well as the ability to aggregate components and the ability to design parametrically with specific values. The translations between parameters he recognizes are: component geometry assembled into device form and component level behaviour is combined to model device behaviour.

Rinderle illustrates his work with the example of a brushless DC motor. The problem is to reduce the standard motor to fit a space and keep length and torque and inside diameter and minimize weight. In this type of domain there is difficulty because of highly-integrated, tightly coupled components. He sees that size, complexity non-linearity of systems and abstraction of relations are difficult for humans to deal with. He also acknowledges that there is predominantly a one-to-many mapping between components and function and that even simple changes are puzzling because of the large influence that they may have on their containing structure. Probably most importantly is the absence of strategies to assist a designer in the recognition of physically significant relations dominating behaviour.

The approach he uses to make the transitions includes optimization techniques to model a system by abstracting specific relations between parameters by moving through design space between optimal points along a plot of the parameters. To identify sets of constraints active at each point he models behaviour as bond graphs.

Within the context of the previously mentioned environment, a designer would adopt the following design method:

- Model the behavioural requirements of the design and the behavioural characteristics of the components with bond graphs.
- Model the geometrical and topological characteristics using an augmented topology graph.
- Link the two of these together and convert them into a physical description of an artifact.

For the conceptual design process Taura & Yoshikawa (1991) identify four stages:

1. Relating of component derived function descriptions onto behaviourally derived functions descriptions,
2. Relating of components to component derived functionality,
3. Relating attributes to components and
4. Relating behaviourally derived functionality to the attributes (Figure 12).

The above corresponds to a functional decomposition, followed by component search, component composition and then evaluation.

In an explanation of an algorithm for generating design solutions, functionality can be extracted from component collections. The crux of the matter is that attributes and behaviour can be inferred from components, functionality cannot. However, Taura & Yoshikawa believe functionality can be determined from knowledge of component behaviours. This is followed by the selection of hypotheses to search for design solutions. An initial hypothesis comprises a set of components. If changed and verified during the design process, this forms a new hypothesis. A hypothesis is changed by altering its related functionality. Finally, there is a mechanism for determining the similarity between hypotheses by performing calculations within component related function descriptions.

The aim of Ulrich and Seering's (1989) research is to develop designs from a functional specification. Central to their work is the use of *Schematic Synthesis*—the generation of a schematic description, or graph of functional elements, in response to a specification of desired behaviour. Essentially there are two steps to this: (i) generate the schematic description; and then (ii) generate a physical description that approximates to the ideal properties of the schematic description. Central to this method is that it involves the decoupling of functional from physical knowledge. The class of design problem tackled includes systems that are described in terms of networks for lumped-parameterized elements, with behaviour that can be specified in terms of the relationship between a single input and a single output. Each network element has associated with it an effort variable and a flow variable (in a similar manner to bond graphs).

For comparison between schematic description and the design specification, the behaviour is abstracted from the descriptions. This is achieved by characterizing the equations of motion for the schema using a three step method: (i) impose constraints from Kirchoff's law; (ii) impose constraints from elemental constitution laws; (iii) from these equations eliminate all of the flow and effort variables except those corresponding to the required input and output variables. Next the equations of motion should be characterized. This means that the relationship between the input and output should be identified.

There are three stages to specifying a problem: (i) specification of the variables of interest by identifying the effort or flow variable of interest; (ii) specification of the relationships of interest by the integration of appropriate derivatives or integrals of the input-output; and (iii) specification of the schematic description corresponding to the dynamic input-output environment.

For solving problems there are also three stages: (i) the generation of candidate schematic descriptions whereby any descriptions that perform a transformation from input quantity to output quantity are selected to form a continuous spine between input and output; (ii) the derivation and classification of the behaviour of these descriptions done by first deriving the equations of motion and then categorizing the equations according to type and number; and (iii) the modification of the candidates through the use of a compact representation (created by the generation of a generalized equivalent description and then the application of some rewrite rules). Furthermore modifications are brought about by the application of domain knowledge.

Despite the many different presentations given here, it is possible to determine a pattern of a process which moves from a required function to a structure through an understanding, or a mapping from required function, to required behaviour, to proposed structure, to evaluations of predicted behaviour, iterating until a satisfactory performance which is considered to be able to satisfy the required function is obtained. Major variations in this pattern are concerned with the "closeness" of function to behaviour, and the type of structure synthesis which is appropriate. In a close function to behaviour step, either because of the nature of the artifact or because of the extent of dependence on past designs, the step can be performed by mapping. Where they are remote it is less likely that function as purpose is meaningfully represented in a system. Structure variations also depend on the extent to which previous components can be re-used, but also depend on the characteristics which need to be represented. Form, for example can be significant in some applications, but not in others.

Where large amounts of re-usable design information, automation of the design process becomes attractive; in other situations the key approach is more likely to depend on flexible modelling and behaviour prediction, leaving the function/structure mapping to the designer.

8 Summary of approaches

This review has examined the modelling of functionality in computer aided design systems in terms of six different issues. Table 4 provides a summary of how each of these issues has been addressed by the key authors or groups of authors. The central issues from the review are discussed and summarized, and for each issue the variation or consensus, if any, between authors is outlined. A theme which has had a significant influence on approaches is the intended application area. Table 4 shows the selected application areas. The relationship between approaches and application is drawn out. Finally, conclusions derived from the review are made.

8.1 The perception of functionality

The two interpretations of function, purpose and action, have been identified. Function as purpose has prime significance in design; yet it is difficult to represent in any way which supports design reasoning or could be used even for design evaluation. Despite the drawbacks, this usage of purpose reflects the nature of the knowledge designers initially utilize in the design process, so its importance cannot be understated. By looking at the nature of artifacts being designed it has been possible to distinguish between those who see function as being distinct from behaviour from those who see external behaviour as synonymous with function. An alternative view of functionality is as action in which case emphasis is placed on information about the mode of action of the design. The importance of this mode of action knowledge becomes apparent when making the transition from

Table 4. Comparative matrix for encapsulating function.

Authors/Issues	Perception	Representation	Manipulation	Framework relations	Design process models	Application
Chakrabarti	purpose	input–output grammar	combination support	physical variable & effects, functions behaviour, structure physical connections	function > structure > behaviour > support-function	mechanisms/ combustion engines
Eastman	purpose	constraint based	combination decomp generalise/class	physical objects functional entities	—	architecture/ building design
Elm	action	input–output	combination support	process components	—	flow processes; light bulb, plant
Faltings	action	qualitative grammar	—	specifications artifacts	spec > function object > function	mechanisms
Finger and Rinderle	purpose	bond graphs	integration & combination	behaviour components	function > behaviour > compositions	mechanical/ gears
Forbus	—	qualitative process-based	—	process behaviour	—	generic processes
Freeman and Newell	purpose	input–output	combination support	requires–provides	function > requirement structure > provides	software engineering
Gero et al.	purpose	labels	—	behaviour structure	function > behaviour structure > behaviour	architecture/ building design
Goel and Chandrasekeran	action	input–output	combination classification	behaviour component/subst	—	physical devices/ processes
Rassmussen	purpose	—	combination support	purpose, abstract, general, & physical functions, form	purpose > function > form	control systems
Rodenacker	purpose	input–output	combination decomposition	process form	function > process > form	mechanisms
Roth	purpose	input–output	combination decomposition	process form	purpose > function > form >	mechanisms
Simon	purpose	—	—	form environment	—	general
Sturges	action	verb–noun labels	decomposition	how–why	function > allocation > component	complex mechanisms
Yoshikawa	action	set theory	combination decomposition set-operations	behaviour form components	function decomposition > component search > comp' composition > evaluation >	mechatronic
Ulrich and Seering	purpose	input–output	combination classification	function schema components	power spine > derive behaviour adaptation	mechanisms
Umeda et al.	purpose	state transformation	decomposition	behaviour	function > behaviour > component > behaviour > match behaviours	mechatronic

function to structure, since most applications assume some kind of direct mapping between the two. The interpretation of function as action seems to be most common with applications which can be considered to be systems with causal interactions. Since there is apparent ambiguity in the use of *function*, the terms function or purpose, and functionality or mode of action are used.

8.2 The representation of functionality

Each of the four representations dealt with has its own advantages and drawbacks (Table 2). A functional grammar has a natural analogue in everyday communication, but suffers from the disadvantage of poor semantics. It has been suggested that with a standard syntax or grammatical language, problems can be overcome. The use of a limited grammar combined with a formalized syntax could enable effective, widely applicable usage. The most likely applications are those for which interpretation of functions as purpose are appropriate.

The input/output representation enables definition of function as purpose only if an external behaviour definition is adopted. It also supports a definition of function as action. Through the input/output relationships, model synthesis can be defined, and top-down design is supported. However, it has the disadvantage of being restricted to systems that incorporate some aspect of flow (of energy, material or information).

Qualitative physics (QP) representation do not appear to readily support synthesis. Instead, they are developed for the analysis of previously constructed systems although the component centred ontology gives some facility for synthesis. A major drawback of QP representations is that a great deal of information is required to predict the performance of a system and once a set of possible behaviours is produced it is difficult to discern between those that are feasible and those that are not. However, some aspects of QP have been utilized, namely the concept of class wide assumptions and the "no-function-in-structure" principle. Of particular note is the use of teleological analysis.

Bond graphs display many of the same advantages and disadvantages as input/output representation. In essence, this is also an input/output representation modelling through creating an analogy of a system's mode of action. However, bond graphs have the additional and important advantages that they relate well to the physical structure of the design and give a strong rigour for modelling synthesis. They provide a useful set of primitives and have a clear indication of the causal relationships within a system. A drawback is that there is no explicit information assisting with the task of abstraction or detailing of a model. More significantly, functionality is not expressed explicitly. It can only be deduced from the behaviours described by a bond graph model.

8.3 The manipulation of functionality

The review of the manipulation of functionality was concerned with methods for operating on representations of function. It addressed combination, breakdown and classification. Both combination and breakdown of functionality are seen as essential to the design process especially during the conceptual phase of the design process. For some representations this is not supported by making the relevant information explicit. Those systems developed to support reasoning about function (Freeman & Newell, 1971), as opposed to those that only support the description of design, generally provide better assistance with manipulation of functionality.

Relevant knowledge to support function combination may be described as statements of mapping relations. Similarly, knowledge to support function breakdown may be held as relationships such as functional decomposition structures, product breakdown structures and mappings. An approach depending upon previously stated cases and relations is useful if the appropriate case exists or one can be adapted with minimal changes. However, storage of a large enough selection of cases is not always feasible. Instead a rule based approach to determine how the knowledge can be manipulated can be adopted.

In the *Functional Decomposition Method* (FDM), Elm (1988) uses a support relationship. In this type of commonly used relationship a supporting function temporally precedes the supported function. Thus using this relationship between functions cannot be strictly regarded as a functional decomposition into a "part of" hierarchy. The outcome of such a decomposition does not produce a set of minor functions exhibiting the same overall functionality as their parent function. Rather, the outcome can be regarded as an example of causality linked chain of function. In contrast, a supporting function is required to enable another function. For example, when using FDM the function "supply heat" would be decomposed FDM into "provide fuel" and "combust fuel". Only the combustion sub-function creates the heat unlike supply fuel which does not.

This is similar to Freeman and Newell *requires-provides* relationship (Freeman & Newell, 1971):

```
function(Input_b, Output_c):–
        function(Input_a, Output_b).
```

8.4 *Design framework relationships*

The relationship between required function and required behaviour is complex and seen by many to be subjective (Umeda et al., 1990). Freeman and Newell (1971) provide a thorough discussion about this relationship. It can be a "one-to-one" relationship but this is rare. Predominantly it is "one-to-many", that is, a single function can be fulfilled by many behaviours.

Not all the attributed behaviours for a given context contribute to a function. In the cases where they do the purpose can be provided for. Those that do not are referred to as "side-effects". A central issue is how the designer makes a translation from function to behaviour. There are a number of main approaches to this:

- *Pragmatic/deductive*—experience of seeing a mechanism under previous similar/identical conditions, allows the deduction that the same will occur again.
- *Empirical*—by experiment the designer can predict the likely behaviours of a design that fulfils a required function.
- *Theoretical*—there is some consensus that the designation of function to a fabricated artifact is subjective. Recent work including qualitative physics, teleology and causal reasoning have made some attempt at tackling the relationship through this approach.

Many authors have contested the notion of mapping function to directly to structure. There is a generally accepted rule by de Kleer and Brown (1984) of *no-function-in-structure*. This adheres to the observation that unless a structure is placed within its operational context then it is not possible to deduce that it fulfils its purpose.

The reverse transition, structure-function, is not directly meaningful since usage of an artifact cannot be determined as stated above. However, mode of action provides an explanation of the design behaviours given the structure and the design context, and therefore forms a basis for anticipating purpose. Transitions between the purpose and mode of action can be associated with transitions between mode of action and structure. Together, these transitions comprise the purpose—structure transition.

8.5 *Models of the design process*

From the study of models of the design process a generalization can be made that design is seen as the transition from a statement of function/purpose to a specification of components/structure. In contrast, the route taken to generate a component set from function requirements varies between workers. Many advocate the manipulation of functionality followed by a phase of mapping to components and then component synthesis. However, the mechanism for making the function component transition shows other variations. The use of knowledge of behaviours is utilized in some cases, as it is suggested knowledge about a direct transition from behaviour to structure is not feasible. Instead, relationships are set up between function and behaviour. This allows mapping of structure to behaviour, a less intuitive process than the direct mapping.

Apart from the order of the tasks within the models for the design process, there are a number of features of interest found within individual design process models. Briefly they include:

- The use of a "requires/provides" relationship to link individual functions at the same level of abstraction.
- The need to express the initial design requirements fully, including a goal specification, performance specification and imposing constraints.
- The need to have a facility for design validation
- The representation of design behaviour without imposing pre-defined structures on subsequent physical realizations of the design.
- The use of cause/effect relations between individual functions of a single design at the same level of abstraction.
- Abstraction requires support allowing for models to have varying levels of applicability; from high-level (generally applicable) to low-level (specific)
- The internal environment needs to be appropriate to the external environment and hence the conditions of operation for a set of process to fulfil a specified purpose should be stated.

8.6 Applications

As a conclusion to this paper it is valuable to highlight the relationship between the applications of interest to different workers and the approaches adopted. From Table 4, it can be seen that applications vary from architecture including building design through mechanical engineering, particularly mechanism design, to process design. Other applications of interest include control systems design and software design. The review therefore covers a wide variety of design experiences.

The pattern which emerges from an examination of the table is that the interpretation of function as purpose most commonly occurs in the design of "static" artifacts for which purpose is not evident from behaviour or from output. In these cases, the concept of function or purpose is a conceptual requirement remote from any real world object which may be applied to meet the function. In contrast, function as action occurs in the design of objects where the purpose is equivalent to some behaviour or the production of some output.

It is hardly surprising from this that representations using input/output, bond graphs, and qualitative physics are the key techniques for these same applications whereas the more abstract function grammar is selected for describing purpose. It similarly follows that the methods for reasoning about function are better established for action than for purpose. In particular, it is evident that most progress has been made using input/output representations applied to systems where there are distinct causal dependencies, either through transformation processes or through mechanical links.

The framework relations tend to highlight the differences already apparent in the interpretations of function and in the varying practices of different design areas. In particular, the arguments about the correctness of having function-structure or behaviour-structure relationships derive from an interpretation of function in the design process. In practice, few have fully explored the complexity inter-relationships in the complete design loop, i.e. required function –⟩ required behaviour in context –⟩ structure ⟨–⟩ mode of action –⟩ attributed behaviours in context –⟩ relevant behaviours + side effects –⟩ potential function. Instead, emphasis has been placed on specific mappings related to function, behaviour and structure with a view to system building. This emphasis is reflected in the corresponding design process models.

Thus it is that the tractability of function as action and the corresponding availability of input/output modelling with associated reasoning has led to design process models with their frameworks being particularly suited to certain classes of design. A clear outcome from this study, therefore, is that there are many areas of design for which handling of function is poorly developed. The more innovative the design, the more conceptual it is; the more new technologies will be introduced, then the more improved support will be required.

Considerable progress has been made in formalizing concepts of function within CAD systems. What has been produced has in many cases provided a formal basis for introducing a much richer base of design and design process knowledge within computer based design support systems. However, much still remains to be done.

Acknowledgements

Funding for this project has been provided by SERC initially through a Research Studentship, and subsequently through the Engineering Design Research Centre, Glasgow, UK. Acknowledgement is made to; Professor David Harland (EDRC-UK) and Dr Bernard Capaldi (formerly EDRC-UK) for their support, Professor A Westerburg of EDRC-US who supervised a stage of this work, and to all those who have contributed through the many discussions which took place during this work.

References

Bobrow, D, 1985, *Qualitative Reasoning About Physical System.* MIT Press.
Cassirer, E, 1923, *Science and Function.* Open Court Publishing/Dover.
Chakrabarti, A, 1989, *Functional Synthesis in Mechanical Conceptual Design.* Technical report, Cambridge University Engineering Department, Internal Report.
Chakrabarti, A, Bligh, T and Holden, T, 1992, "Towards a decision-support framework for the embodiment-phase of mechanical design". *Artificial Intelligence in Engineering* **7** 21–36.
Chandrasekaran, B, 1989, "A framework for design problem solving". *Research in Engineering Design* **1** 75–86.
Davidson, G, Seaton, M, Simpson, J and Kirkpatrick, E (eds) 1985, *Chamber Concise 20th Century Dictionary.* W & R Chambers Ltd.
de Kleer, J, 1979, *Causal and Teleological Reasoning in Circuit Recognition.* PhD thesis, MIT Artificial Intelligence Laboratory Technical Report 529.
de Kleer, J, 1984, "How circuits work". *Artificial Intelligence,* **24**(3) 205–280.
de Kleer, J and Brown, J, 1984, "A qualitative physics based on confluences". *Artificial Intelligence* **24.**
de Kleer, J and Brown, J, 1986, "Theories of causal ordering". *Artificial Intelligence* **29** 33–61.
Eastman, C, 1982, "Recent developments in representation in the science of design". In: *IEEE, Design Automation Conference Proceedings,* Nashville, TN, 29 June–1 July.
Eastman, C, 1987, "Fundamental problems in the development of computer based architectural design models". In: *Design Abstractions.*
Eastman, C, Bond, A and Chase, S, 1989, *A Formal Approach for Product Model Information.* Research Report 3, Graduate School of Architecture and Urban Planning, University of California, November.
Elm, W, 1988, *Functional Structure Methodology.* Westinghouse Electrical Corporation (unpublished).
Ernst, C and Newell, A, 1969, *GPS: A Case Study in Generality and Problem Solving.* Academic Press.
Faltings, B, 1991, "Qualitative models in conceptual design". In: *Proceedings of the first Conference on Artificial Intelligence in Design,* pp 645–663.
Finger, S and Rinderle, J, 1989a, *Transforming Behavioural and Physical Representations of Mechanical Designs.* Technical report, Carnegie-Mellon University.
Finger, S and Rinderle, J, 1989b, "A transformational approach to mechanical design using a bond graph grammar". In: *First ASME Design Theory and Methodology Conference,* Montreal, Quebec, September.
Forbus, K, 1984, "Qualitative process theory". *Artificial Intelligence* **24.**
Forbus, K, 1988a, "Intelligent computer aided engineering design". *AI Magazine,* Fall, 23–36.
Forbus, K, 1988b, "Qualitative physics: Past, future and present". In: H Schrobe (ed.) *Exploring Artificial Intelligence: A Survey on talks from the national conferences on Artificial Intelligence,* pp 239–296, Morgan-Kauffman.
Forbus, K, 1988c, "QPE: Using assumption-based truth maintenance for qualitative simulation". *Artificial Intelligence in Engineering.* **3**(4) 200–215.
Fowler, H and Fowler, F, (eds.), 1984, *The Oxford English Dictionary (7th Ed.).* Oxford University Press.

Freeman, P and Newell, A, 1971, "A model for functional reasoning in design". In: *Proceedings IJCAI*, pp 621–639.

Gawthrop, P, 1989, "Symbolic generation of robot of simulations". In: *Third European Simulation Congress*, Edinburgh.

Gawthrop, P and Smith, L, 1990, "An environment for the specification, design, operation, maintenance and revision of manufacturing control systems". In: *UKIT 1990 Conference*, Southampton, UK.

Gawthrop, P and Smith, L, 1992, *MetaModelling*. Manuscript in preparation, Control Group, Department of Mechanical Engineering, University of Glasgow.

Gawthrop, P, Marrison, N and Smith, L, 1990, *MTT: A Bond Graph Toolbox*. Technical report, Control Group, The University of Glasgow.

Gero, J, Lee, H and Tham, K, 1991, "Behaviour: A link between function and structure in design". In: *IFIP W.G. 5.2. Workshop on Intelligent CAD*, Columbus, OH, 30 September–3 October.

Goel, A, 1991, "Representation of design functions in experience-based design". In: *IFIP W.G. 5.2 Workshop on Intelligent CAD*, Columbus, OH, 30 September–3 October.

Goel, A and Chandrasekaran, B, 1989, "Functional representation of designs and redesign problem solving". In: *Proceedings Eleventh International conference on Artificial Intelligence*, Detroit, MI, August.

Hoover, S and Rinderle, J, 1989, "A synthesis strategy for engineering devices". Submitted to *Research in Engineering Designs*.

Hundal, M, 1988, "Methods for systematic development of function structures with application to computers". In: *Proceedings of the International Conference on Engineering Design, ICED '88*, pp 197–206, Budapest.

Hundal, M, 1990, "Research in design theory and methodology in West Germany". In: *Proceedings of the 2nd International Conference on Design Theory and Methodology*, pp 235–238, American Society of Mechanical Engineers, September.

Imamura, S, Kojima, T and Sekiguchi, H, 1988, "A study on the object oriented product model–Representation of Geometry and dimension". *Annals of the CIRP* **37** 127–130.

Iwasaki, Y and Simon, H, 1988, "Causality in device behaviour". *Artificial Intelligence* **29** 3–32.

Iwasaki, Y and Simon H, 1986b, "Theories of causal ordering: A reply to de Kleer and Brown". *Artificial Intelligence* **29** 63–72.

Joskowicz, L, 1988a, *Mechanism Comparison and Classification for Redesign*. Technical Report RC 14573 (65152), IBM Research Division, TJ Watson Research Center, NY.

Joskowicz, L, 1988b, "Reasoning about the kinamatics of mechanical devices". *Artificial Intelligence in Engineering*.

Joskowicz, L and Addanki, S, 1988, *Innovative Design of Kinematic Pairs*. Technical Report RC 14506 (64962), IBM Research Division, TJ Watson Research Center, NY.

Karnopp, D and Rosenberg, R, 1975, *System Dynamics: A Unified Approach*. Wiley.

Kuipers, B, 1984, "Commonsense reasoning about causality: Deriving behaviour from structure". *Artificial Intelligence* **24** 169–203.

Kuipers, B, 1986, "Qualitative simulation." *Artificial Intelligence*. **29** 289–338.

Lai, K, 1987, "Mechanical design simplification using function description language". In:*Proceedings of the 15th North American Manufacturing Research Conference (NAMRC)*, Bethlehem, PA, May 27–29.

MacCallum, K, Duffy, A and Green, S, 1987, "Intelligent concept design assistant". In:*Design Theory for CAD: IFIP WG 5.2 Working Conference*. North-Holland.

Mitchell, W, 1989, *The Logic of Architecture*. MIT, Press.

Mitchell, T, Steinberg, L and Shulman, J, 1985, *A Knowledge-Based Approach to Design*. Technical Report LCSR-TR-65, Computer Science Department, Rutgers University, January.

Pahl, G and Beitz, W, 1984, *Engineering Design*. Springer-Verlag.

Paynter, H, 1961, *Analysis and Design of Engineering Systems*. MIT Press.

Rasmussen, J, 1984, *On Information Processing and Human-Machine Interaction, An Approach to Cognitive Engineering*. Elsevier.

Rinderle, J, Colburn, E, Hoover, S, Paz-Souldon, J and Watton, J, 1989, "Form-function characteristics of electro-mechanical designs". In: *Design Theory '88: Proceedings of the 1988 NSF Grantee Workshop on Design Theory and Methodology*. Springer-Verlag.

Rodenacker, W, 1976, *Methodisches Konstruieren*. Springer.

Roth, K, 1982, *Konstruieren mit Konstruktions Katalogen*. Springer.Verlag.

Roth, K, Franke, H and Simonek, R, 1971, "Algorithmischeses Auswahlverfahren zur Konstruktion mit Katalogen". *Feinwerktechnik* **75** 337–345.

Simon H, 1952, "On the definition of the causal relation". *Journal of Philosophy* **49** 517–528.

Simon, J, 1982, *The Sciences of the Artificial*. MIT Press.

Smith J and Smith, D, 1972, "Data-base abstractions: Generalizations and abstractions". *ACM Transactions on Database Systems* **2**(2) 105–133.

Sturges, R, 1981, *The Function of Value Engineering*. Technical report, Department of Mechanical Engineering, Carnegie-Mellon University.

Sturges, R, Kilani, M and O'Shaughnessy, K, 1991, *An Extension of Function Logic Representations for Preliminary Design*. Technical report, Department of Mechanical Engineering, Carnegie-Mellon University, April.

Taura, T and Yoshikawa, H, 1991, "A metric space for intelligent CAD". In: *IFIP W.G. 5.2 Workshop on Intelligent CAD*. Columbus, OH, 30 September–3 October.

Ulrich, T and Seering, W, 1989, "Synthesis of schematic descriptions in mechanical design". *Research in Engineering Design* **1** 3–18.

Umeda, Y, Takeda, H, Tomiyama, T and Yoshikawa, H, 1990, "Function, behaviour and structure". In: *Proceedings of AI in Engineering '90*, Boston, MA.

Urdang, L, 1989, *Synonym Dictionary*. Longman.

Weld, D and de Kleer, J, 1990, *Readings in Qualitative Reasoning About Physical Systems*. Morgan Kaufmann.

Williams, B, 1984, "Qualitative analysis of MOS circuits". *Artificial Intelligence* **24**.

Williams, B, 1991, "Interaction-based design: Constructing novel devices from first principles". In: *IFIP W.G. 5.2 Workshop on Intelligent CAD*, Columbus, OH.

Yoshikawa, H, Bo, K and Lillehagen, F, (eds.), "CAD framework guided by general design theory". In: *CAD Systems Framework*. pp 241–344, North-Holland.

Appendix A Extended bibliography

Akman, V, ten Hagen, P, Rogier, J and Veerkemp, P, 1987, "Knowledge engineering in design". Technical report, Department of Interactive System, Center for Mathematics and Information Science (CWI), Kruislaan 413, 1098 SJ Amsterdam, Netherlands.

Andreasen, M, 1985, "Methodology for layout design ('ENTWERFEN')". In: *Proceedings of the International Conference on Engineering Design, ICED*, 257–267, Hamburg, 26-28 August.

Andreasen, M, 1990, "Strengthening the DFA rules by dispositional mechanisms". In: *11th International Conference on Assembly Automation*, Dearborn, MI, 11–14 November.

Bandekar, V, 1989, "Causal models for diagnostic reasoning". *Artificial Intelligence in Engineering* **4**(2).

Blundell, J, 1990, "Functional design". In: *Proceedings of Manufacturing International '90, Part 2: Advances in Manufacturing Systems,* Atlanta, GA, 25–28 March.

Bobrow, D, 1985, *Qualitative Reasoning About Physical System*. MIT, Press.

Bredeweg, B and Wielinga, B, 1988, "Integrating qualitative reasoning approaches". In: *Proceedings of ECAI-88,* Munich, Germany.

Bylander, T, 1988, "A critique of qualitative simulation from a consolidation viewpoint", *Transactions on Systems, Man and Cybernetics* **18**(2) March/April.

Bytheway, C, 1985, "Basic function determination techniques". In: *Proceedings of Fifth National Meeting, volume 11*. Society of American Value Engineers.

Cross, N, 1989, *Engineering design methods*. Wiley.

Davis, R, 1984, "Diagnostic reasoning based on structure and behaviour". *Artificial Intelligence* **24** 347–410.

Dixon, J, 1986, "Artificial intelligence and design: A mechanical engineering view". In: *Engineering Applications*.

Dormoy, J and Raiman, O, 1990, "Assembling a device". *Artificial Intelligence in Engineering* **3**(4) 216–226.

Duhovnik, J, 1990, "Function classification of structure elements in CAD". In: *International Conference on Engineering Design, ICED '90*, 810–821, Dubrovnik, 28-31 August.

Eastman, C, 1990, *Explorations of the Cognitive Process in Design*. Technical Report AD671158, Computer Science Department, Carnegie-Mellon University, Pittsburgh, PA.

Eswaran, K and Chamberlin, D, 1975, "Functional specifications of a subsystem for data base entry". In: *Proceedings of the Conference on Very Large Databases,* ACM.

Fenves, S and Baker, N, 1988, "Spatial and functional representation language for structural design". In: *IFIP WG 5.2, Expert Systems in Computer-Aided Design*. Elsevier, 19–22 September.

Finger, S and Dixon, J, 1989, "A review of research in mechanical engineering design. Part II: Representations, analysis, and design for life cycle". *Research in Engineering Design* **1** 121–137.

Finger, S and Rinderle, J, 1989, *Representation of Mechanical Designs*. Technical report, Carnegie-Mellon University.

Forbus, K, 1981, "Qualitative reasoning about physical process". In: *Proceedings of IJCAI-81*. Vancouver, BC, Morgan Kaufmann.

French, M, 1985, *Conceptual Design for Engineers*. Design Council, London.

French, M, 1990, "Function costing: A potential aid to designers". *Journal of Engineering Design* **1**(1).

Grabowski, H and Seiler, W, 1985, "Techniques, operations and models for functional and preliminary design phases". In: H Yoshikawa (ed.), *Design and Synthesis* 17–22, Elsevier.

Grinman, M, Nishnevich, V and Rokhlin, V, 1990, "Functional designing of the future of the turbine plants". In: *International Conference on Engineering Design, ICED '90*, 245–250, Dubrovnik, 28–31 August.

Gui, J, 1990, "Function-behaviour-structure machine design model and its use in assembly sequence planning". *Journal of Engineering Design.* **1**(3).

Hauck, P and Taylor, D, 1991, "Deriving physical design constraints from functional topologies". *Research in Engineering Design* **2** 81–92.

Hayes, P, 1979, "The naive physics manifesto". In: D Michie (ed.), *Expert systems in the Micro-Electronic Age* pp 242–269, Edinburgh University Press.

Hayes, P, 1945, "The second naive physics manifesto". In: J Hobbs and R Moore (ed.), *Formal Theories of the Commonsense world* 468–485, Ablex.

Horvath, I, 1990, "Selection and combination of part segments by function". In: *International Conference on Engineering Design, ICED '90,* pp 91–98, Dubrovnik, 28–31 August.

Hubka, V, 1982, *Principles of Engineering Design.* Butterworth Scientific.

Hubka, V, 1985, "Attempts and possibilities for rationalisation of engineering design". In: H Yoshikawa (ed.), *Design and Synthesis* pp 133–138, Elsevier.

Hubka, V, 1989, "A scientific approach to engineering design". *Design Studies* **8**(3), July.

Hundal, M, 1990, "Research in design theory and methodology in West Germany". In: *Proceedings of the 2nd International Conference on Design Theory and Methodology.* pp 235–238, American Society of Mechanical Engineers, September.

Hundal, M and Byrne, J, 1990. "Computer-aided generation of function block diagrams in a methodological design procedure". In: *Proceedings of the 2nd International Conference on Design Theory and Methodology,* pp 251–257, American Society of Mechanical Engineers, September.

Iwasaki, Y, 1988, *Model Based Reasoning of Device Behaviour with Causal Ordering.* PhD thesis, Department of Computer Science, Carnegie-Mellon University.

Iwasaki, Y and Simon, H, 1986, "Theories of causal ordering; A reply to de Kleer and Brown". *Artificial Intelligence* **29** 63–72.

Johnson, A, 1988, *Functional modelling: A new development in computer aided design.* Technical report, Department of Engineering, University of Cambridge.

Johnson, A, 1991, "Designing by functions". *Design Studies* **12**(1); 51–56.

Kota, S and Lee, CL, 1990, "Functional framework for hydraulic systems design using abstraction/decomposition hierarchies". In: *Proceedings of ASME International Computers in Engineering,* New York.

Kota, S and Ward, A, 1990, "Functions, structures, and constraints in conceptual design". In: *Proceedings of the 2nd International Conference on Design Theory and Methodology,* pp 239–250, American Society of Mechanical Engineers, September.

Kuipers, B and Berleant, D, 1988, "Using incomplete qualitative knowledge in qualitative reasoning". In: *Proceedings of the National Conference on Artificial Intelligence,* pp 324–329, American Association for Artificial Intelligence.

Kuipers, B, de Kleer, J and Brown, J, 1981, *Mental Models; A Critic.* Working Papers in Cognitive Science 17, Tufts University, Medford, MA.

Lai, K and Wilson, W, 1987, "FDL: A language for functional description and rationalisation in mechanical design". In: *Computers in Engineering, ASME,* pp 87–94, New York.

Margolis, D and Karnopp, D, 1979, "Bond graphs for flexible multibody systems". *Transactions of the ASME* **101**; 50–57.

Nayak, P, Joskowicz, L and Addanki, S, 1991, "Context-dependent behaviors: A preliminary report". In: *IFIP W.G. 5.2 Workshop on Intelligent CAD.* Columbus, OH, 30 September–3 October.

Novak, G, 1976, *Computer Understanding of Physics Problems Stated in National Language.* PhD thesis, Department of Computer Science, University of Texas at Austin.

Popplestone, R, 1985, "The applications of artificial intelligence techniques to design systems". In: H, Yoshikawa (ed.), *Design and Synthesis,* pp 453–458. Elsevier.

Popplestone, R, 1987, "The Edinburgh designer system as a framework for robotics: The design of behaviour". *AI EDAM, 25–36.*

Pu, P, 1989, *Qualitative Simulation of Ordinary and Intermittent Mechanisms.* PhD thesis, University of Pennsylvania.

Pugh, S, 1985. "CAD/CAM–Its effect on design understanding and progress". In: *International Conference on CAD/CAM, Robotics and Automation,* Tuscon, February.

Pugh, S and Morley, I, 1988, *Total Design.* Design Division, University of Strathclyde.

Pugh, S and Smith, D, 1976, "CAD in the context of engineering design–the designers viewpoint". In: *CAD 76 Proceedings* pp 193–198.

Rinderle, J, 1982, *Measures of Functional Coupling in Design*. PhD thesis, Massachusetts Institute of Technology.

Rinderle, J and Suh, N, 1989, "Measures of functional coupling in design". In: *Journal of Engineering for Industry* pp 383–388, November.

Rodenacker, W and Baumgarth, R, 1976, "Die Vereinfachung der Gerate beginnt mit der Funktionsstruktur (The Simplification of Devices Begins with the Function Structure". *Konstruktion* **28** 479–482.

Roth, K, 1985, "Modellbildung fur das methodische Konstruieren ohne und mit Rechnerunterstutzung (Model Formation for Methodical Designing without and with Computer Support". In: *International Conference on Engineering Design. ICED '85* Hamburg, pp 633–644.

Roy, U and Liu, C, 1988, "Establishment of functional relationships between product components in assembly database". *CAD* **21** 561–569.

Semekel, H, 1989, "Functional models and design solutions". *Annals of the CIRP*. January.

Sembugamoorthy, V and Chandrasekaran, B, 1984, *Functional Representation of Devices and Compilation of Diagnostic Problem-Solving Systems*. Technical report, AI Group, Ohio State University.

Shrager, J, 1988, "Issues and direction in qualitative physics: Special edition from the 1987 workshops". *Artificial Intelligence in Engineering* **3**(3) 122–124.

Siewiorek, D, Giuse, D, Birmingham, W, Hirsch, M, Rao, V and York, G, 1982, *DEMETER Project: Phase 1 (1984)*. Technical report, Carnegie-Mellon University. University of London.

Smithers, T, Conkie, A, Doheny, J, Logan, B and Millington, K, 1989, "Design as intelligent behaviour: An AI in design research programme". In: *Fourth International Conference on Applications of Artificial Intelligence in Engineering,* Cambridge. UK, 11–14 July.

Sriram, D, Stephanopoulos, G, Logcher, R, Gossard, D, Grolean, N, Serrano, D and Navinchandro, D, 1989, "Knowledge-based system applications in engineering design: Research at MIT". *AI Magazine*. **10**(3) 79–95.

Stadlbauer, H, 1991, *Functional-Skeletons: From Specification to Design*. Technical report, Institute of Flexible Automation, Technology University of Vienna, Austria.

Stadlbauer, H, 1991, *Functionality-Oriented Geometric Modelling*. Technical report, Institute of Flexible Automation, Technology University of Vienna, Austria.

Stadlbauer, H, 1991, *The Application of Functional-Skeletons to Reduce the Complexity of Finite Element Analysis*. Technical report, Institute of Flexible Automation, Technology University of Vienna.

Steels, L, 1989, *Diagnosis with a Function-Fault Model*. Hemisphere Publishing.

Sticklen, J and Chandrasekaran, B, 1989, "Integrating classification-based compiled level reasoning with function-based deep reasoning." Hemisphere Publishing.

Struss, P, 1987, "Multiple representation of structure and function in an intelligent environment". In: J Gero (ed.), *Expert Systems in Computer-Aided Design*. Elsevier.

Subrahmanian, E, Podnar, G and Westerberg, A, 1990, *A Shared Computational Environment for Concurrent Engineering*. Technical report, Engineering Design Research Center, Carnegie-Mellon University.

Ulrich, K, 1988, *Computational and Pre-Parametric Design*. PhD thesis, MIT Artificial Intelligence Lab.

Ulrich, T and Seering, W, 1988, "Computation and conceptual design". In: *Robotics and Computer-Integrated Manufacturing*. **4**(3/4) 309–315.

Ulrich, T and Seering, W, 1988, "Conceptual design: Synthesis of systems components". In: *Intelligent and Integrated Manufacturing Analysis and Synthesis,* pp 57–66, American Society of Mechanical Engineers, New York.

Ulrich, T and Seering, W, 1988, "Function sharing in mechanical design". In: *Proceedings of the National Conference on Artificial Intelligence*, pp 342–346, American Association for Artificial Intelligence.

Yoshikawa, H, 1981, "General design theory and intelligent behaviour". In: T Sata and E Warman (eds.) *Man-Machine Communication in CAD/CAM,* pp 35–53, North-Holland.

Yoshikawa, H, 1982, "General design theory and a CAD system". In: *Man-Machine Communication in CAD/CAM, Proceedings of IFIP WG 5.2/5.3 Working Conference (Tokyo)*. North-Holland.