
GUEST EDITORIAL

Special Issue: Representing functionality in design

AMARESH CHAKRABARTI AND LUCIENNE BLESSING

Engineering Design Centre, Cambridge University Engineering Department, Trumpington Street,
Cambridge CB2 1PZ, United Kingdom

As designs exist to satisfy some purpose or function, knowledge of functionality is essential in a wide variety of design-related activities, including generation and modification of designs, comparison, evaluation and selection of designs, and explanation, diagnosis or repair of designs. Functional modelling refers to a wide variety of approaches to model a design and its requirements from its functional aspects so as to allow reasoning about its functionality for various activities such as the above.

Function has been historically interpreted in a wide variety of ways: for instance, as an abstraction of the intended behavior of a design, an indexing of its intended behavior, the relationship between a design and its environment, the external behavior of a design, or its internal behavior.

Functional reasoning as a design approach has been around for at least 25 years now (Koller, Rodenacker, and Roth in Germany, French in the United Kingdom, Freeman and Newell in the United States, Hubka in Switzerland, and Yoshikawa in Japan are but a few examples of the early researchers in this area), and have attempted to support design in the conceptual stage by methods and approaches to describe function, to establish function structures (e.g., by using generally valid functions), to satisfy these subfunctions and combine them into concept alternatives (e.g., by using catalogues of physical effects and working principles), and to evaluate these (e.g., by using morphological matrices). However, the emphasis was largely prescriptive, and computer supports passive in nature. The advent of computers and the development of artificial intelligence (AI) techniques provided a renewed focus on reasoning about functions, extending the area into diagnosis and explanation, and allowed computers to take a more active role in the design process, especially in its generative aspects.

A formal representation of functionality is essential for supporting any of these activities on computers. Tradition-

ally, there have been three approaches in representing function:

- Verb–noun pairs, such as “transmit torque”; this representation has been around as long as humans have communicated with others how their designs do, or do not work.
- Input–output flow transformations, where the inputs and outputs can be energy, materials, or information; the origin can be traced to the early works of Weizsacker, Rodenacker, and Koller in Germany.
- Transformation between input–output situations, stemming from the early work by Hubka in Switzerland.

The goal of this special issue is to bring together the state of knowledge of representing functionality in design. The idea grew out of the activities in the Workshop on “Representing Function in Design,” held at the 1994 AI in Design conference in Lausanne, which attracted about thirty top researchers from around the world discussing issues such as what function means, what their specific function representations are, and how these are applicable in solving real-world design problems. A common ground for these discussions was provided by allowing the participants to apply their functional modelling methods to a common real-world design problem.

This issue contains seven papers from five different countries, rigorously reviewed by at least three reviewers from three different countries, all different from the country from which the paper came. Between them, these papers use all three functional representations described before, as well as their blends. A wide variety of tasks are addressed in these papers: description and elaboration of function requirements, generation, evaluation and explanation of conceptual solutions, and diagnosis of solutions when they do not work as expected. The papers show a variety of approaches to generating concepts: composition, adaptation, and retrieval of concepts. They are applicable in a wide variety of

Reprint requests to: Dr. Amaresh Chakrabarti, Engineering Design Centre, Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, United Kingdom.

areas, including mechatronics, mechanisms, architectural design, process design and structural design, and operate at, or encompass a number of abstraction levels.

The papers are organized in this issue according to the part of the design process to which they apply (from task clarification to more detailed phases of design) and in terms of the functional representation adopted (starting from verb–noun pair, through input–output [I–O] flow transformation, to I–O state transformation). The first three papers, by Sturges et al., Umeda et al., and Qian and Gero, are based on verb–noun representation, and deal, respectively, with function structure elaboration, search and visualization of behavioral characteristics of concepts, and analogy-based interactive creative design. The next three papers, by Chakrabarti and Bligh, Bracewell and Sharpe, and Peien et al., are based on I–O flow transformation, and deal, respectively, with automated generation of mechanical design concepts and their spatial configurations, computer support to generation, evaluation and visualization of physical systems, and concept retrieval and evaluation based on design catalogues. The last paper, by Goel and Stroulia, is on diagnosis of faults in physical devices, and uses a behavioral state transformation representation of function. The next few paragraphs give a short summary of each paper.

Sturges et al. define function as the domain-independent characteristics or behavior of elements or groups of elements, and modify function logic methods to develop and use function block diagrams. The idea is that the designer should be able to describe the intended function, expand it into required subfunctions, and map these into components capable of fulfilling them. The design is assisted by the computer in this process in terms of systematic identification of functions, allocation of constraints to each function, the interrelations between functions, and the evaluation of the functions. The approach supports the designer mainly in the identification, articulation, and evaluation of function structures, rather than the search for solutions, and therefore applies to the later stages of task clarification and the earlier stages of conceptual design. The approach is applicable in multiple domains, and has been implemented into a prototype function block diagram generator program. According to the authors, the main benefit of using the software is the promotion of discussion and understanding of the design problem, although no rigorous testing is reported in the paper.

Umeda et al. describe function as behavior abstracted by human to utilize it, and thereby explicitly state function as a subjective indexing of some behavior. The task here is to assist the designer in the synthetic as well as analytic aspects of conceptual design. They set out to do so by providing a computer support within which designers should be able to specify required function, decompose this into subfunctions, embody these functions using appropriate behaviors from a database, and visualize the functioning of the resulting working principles. The computer supports the search for behaviors to satisfy the functions, checks in the resulting information, and proposes modifications to rectify the incon-

sistencies. The approach is based on the Metamodel Concept of Tomiyama and Yoshikawa, and the Qualitative Process theory of Forbus. The approach is implemented into a prototype function–behavior–state modeler, and has been tested by three groups of designers. It has been found applicable at least in mechanical and building designs.

Qian and Gero define function as “to do something” (such as buzzing or opening), and use the relationships between structures, behavior, and function of existing designs across domains to index them so as to retrieve and modify analogous designs to suit them to a given function interactively on computers. The aim is to assist in creative design by analogical reasoning, and the ethos is supporting the designer rather than automation. The approach is to retrieve a target concept, use its function, behavior, and structural indices to retrieve other analogous designs in the database, and transform these designs into a new design, with input from the designer. The concepts dealt with are at a fairly abstract level, and the approach seems to work well in nondynamic structures. The core system is implemented in a program called DESSUA, and the approach explained using several design problems, although no results of testing and evaluation are reported.

Chakrabarti and Bligh use function as a transformation of the characteristics of flow variables, such as the change in orientation of torque. They define a database of basic building blocks in mechanical devices, in terms of the various possible ways of transforming an input into an output. The transformation can be in terms of the kind, orientation, direction, position, and magnitude. Given the required function in terms of a transformation between some I–O characteristics, these building blocks are automatically composed, using constrained search, by the software FuncSION. This results in a set of possible alternative concepts that could satisfy the function in terms a number of possible spatial configurations. The idea is to expand designers’ creativity by offering them a range of possible alternatives and supporting them to explore these at various levels of abstraction. The creativity of the software has been tested by comparing the concepts generated in a number of design case studies with those generated by FuncSION.

The paper by Bracewell and Sharpe uses function as an I–O description based on the bond graph language to support generation of qualitative schemes and their preliminary embodiments. The designer uses the building site of the software to define the required function and tries to satisfy it by either further decomposing the function into subfunctions, or by using possible alternative means to satisfy each function. Based on a modified function–means–tree approach, the system can support design of interdisciplinary, continuous time, energetic systems. The designers are assisted in the generation of schemes by allowing them to consult the databases for working principles, means and components, and in the evaluation and simulation of these schemes. The system emphasizes support rather than automation, and has been implemented in a prototype software

called “Schemebuilder.” The use of the system is explained using a drug infuser design example, although no results of testing and evaluation of the system using real design cases has been provided in the article.

Peien et al. use the I–O flow transformation definition of function. Standard functions, such as “separate materials,” are described as a hierarchy of function detail (such as separate a soft material from a hard one when both are granular), and comprehensive catalogues of existing working principles are also developed. These principles are indexed in terms of a wide variety of parameters. The designer can edit a function hierarchy from the database to specify the required function, and use a host of indexing parameters to specify the requirements to the design. The computer uses these to retrieve possible concept alternatives from the catalogues and to support an overall evaluation of these alternatives. A direct search technique using multiple criteria is used for retrieval, and can be used in any domain as long as the functions are applicable. The general idea is that the designer should be able to describe the intended function, retrieve existing concepts from design catalogues, check for their feasibility, and evaluate them on computers. The work is implemented, and although no results of testing of the software is mentioned explicitly, some examples are provided to demonstrate the approach.

Goel and Stroulia use intended output behavior of a system to define function, where a function is represented in terms of I–O behavioral states. The aim of the project is automated diagnosis of faults in physical devices as part of adaptive and redesign activities. Three kinds of diagnosis tasks are attempted: where a design does not function, where a design exhibits undesired behaviors, and where a specific element misbehaves. The central idea is to model devices in terms of their function, behavior and structure, and link these models with one another such that diagnostic tasks, which typically involve traversing causal connections between these three models, can be attained. For instance, given the struc-

ture, expected behavior and desired function of a device as input to the diagnosis program KRITIK2, it can suggest possible structural faults responsible for the difference in function and behavior. The differences in function are parametric at present (such as not loud enough, or not low enough), and the devices are at a physical-effect-like level of detail. The examples illustrate that the ideas are applicable in a wide range of domains, although no results of any rigorous testing is reported.

A number of trends can be observed from the above papers. It is encouraging to see the wide variety of domains and tasks where functional representations and models are potentially applicable and useful. It is also positive to observe the emergence of consensus toward designer-supports rather than automation of the design process. However, very few of the systems have been tested on real design cases, or using real designers in industrial environments; this issue needs to be addressed seriously. The papers between them encapsulate a multiplicity of views about function, which is not necessarily a problem, as they find their uses in different tasks and domains. However, it would be interesting to ask if there could be a unified view. Another trend is that the papers address mainly the issue of generating concepts to satisfy a required function. There is relatively little work into supporting the clarification of functionality. Evaluation, of alternative formulations of the required functionality as well as of alternative design solutions, has also been, by and large, a neglected area of research that needs substantial research input before an overall functional reasoning support could be developed.

ACKNOWLEDGMENTS

The guest editors gratefully acknowledge the support from all the reviewers, and the valuable editorial support and suggestions from Clive Dym, the editor of *AI EDAM*, as well as from Morrell Gillette of Cambridge University Press.