INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 97 TAMPERE, AUGUST 19-21, 1997

SUPPORTING DEEP UNDERSTANDING AND PROBLEM SOLVING USING FUNCTION STRUCTURES: A CASE STUDY

Amaresh Chakrabarti

function structures, conceptual design, generation, problem identification, functional reasoning

1 Introduction

A function structure is a structured description of the functionality of a system in terms of its functions and their inter-relationships. Being an intermediate description between the design brief and the eventual concept descriptions, it should foster problem understanding and solving in a deep sense so as to help identify the right problem to be solved, and help solve the problem right by fostering synthesis and evaluation of a wide range of concept alternatives so as to fulfil the functional requirements of the problem contained in the design brief.

Design research emphasises the importance of drawing function structures [Roth, 1970; Hubka, 1982; Pahl & Beitz, 1996]; passive supports for function structure development also exist [Sturges et. al., 1993]. However, little guidance is available as to how to develop these, and little empirical evidence that the existing approaches foster deep problem understanding and solving. This paper uses a case study to empirically evaluate a major function structure approach, and reviews several others in the context of this study, to identify the features a function structure approach must have for fostering deep understanding and problem solving.

2 Deep Understanding and Problem Solving: *Depth Metrics*

A deep understanding of the problem should help identify the right problem to be solved. However, it is often hard to decide what the right problem is. Design research provides little guidance in this respect, although Pahl & Beitz [1996] suggests to use abstraction to broaden the problem statement. However, the understanding of the problem could still be enhanced in two ways. One is to identify possible problem areas of the existing solutions: the wider the possible perspectives from which problems of an existing solution are postulated in its use and abuse, the better this understanding should be. The other is to broaden the problem into a variety of abstractions: the wider this variety, the better the understanding should be.

Problem solving in a deep sense depends on a deep understanding of the problem, and should enable generation of not just a wide *variety* of concepts, where *variety* is characterised in terms of the range of different function structures considered by the designer, as well as the range of partial solutions for each function within the function structures. The concepts must also be *rich* in the sense that their partial solutions should work together into a organic whole, rather than a bagful of disjoint ideas; *richness* is characterised by the *completeness* and

consistency of the concepts. While *consistency* characterises how well the overall function, function structures and concepts relate to one another, *completeness* characterises how well connected the partial solutions in a concept are (whether they are: at the same abstraction level, connected meaningfully, or detailed enough for evaluation of the concept's feasibility).

The following *depth metrics* therefore were devised to assess deep understanding and problem solving: (i) an appreciation of the main functions of the problem, and the possible problem areas of existing solutions contained in the design brief; (ii) breadth of the problem perceived; (iii) richness of the concepts in terms of completeness and consistency of the solutions; (iv) the variety of function structures and partial solutions within the solution space considered.

3 A Case Study and Some Observations

The conceptual design approach taught in Cambridge [Wallace, 1996] is based on Pahl and Beitz [1996], and uses *system* (to describe the functions within a design, eg, a lawn mower would have functions such as *cut grass*, *store cuttings* etc) and *process* (to describe how the design functions in its working environment, eg, *position device*, *store device*, etc) function structures. The case study used here is based on a conceptual design exercise undertaken by first year students, already familiar with the conceptual design approach, who are recommended to follow this approach to design concepts to a given problem, and write a report. The design problem set was: "If you wish to cross the English Channel by car through the Channel Tunnel, you must drive your vehicle into "Le Shuttle", the Channel Tunnel Train. Vehicles must enter the rear of the train from the side, as shown in the figure (not included), and then drive towards the front until they approach the vehicle ahead. At this point, an official guides your vehicle forward, by using hand signals, until it is a short distance from the vehicle in front, when you are signalled to stop and put on the brake. This process is repeated for each vehicle in turn. All this is time consuming and requires an official for each train and deck. You are to design a means of loading vehicles into "Le Shuttle" quickly, economically and safely, while ensuring the correct spacing between vehicles. It could be active or passive."

A sample of 50 of 310 reports was analysed to ask the following questions to evaluate if the function structure approach fosters problem understanding and solving in a deep sense:
(i) The extent to which students used function structures in their design process (to help distinguish those that used function structure from those that tried to post-fit them, only to appear to have used them). We feel that a student who did not use function-structures to generate concepts but simply tried to post-fit them into a morphological matrix, would be likely to list the partial solutions for each concept in single, isolated columns of the matrix.
(ii) The quality of the design process, in terms of the *depth metrics*, of those who used function structures (to help adjudge the usefulness of function structures).

A post-design questionnaire survey was conducted among the students to see if they found function structures (and other steps) useful and well-taught; this was to compare their responses to the analysis that it was not unduly biased, and also how they felt about function structures development relative to the other steps they followed during the design process.

We found that a large number (15 among 50) of students did not use function structures but only tried to post-fit solutions into them. The results of the survey supports this, which suggests that students found function structures the least useful (2.34 against an average of

2.83, where 1 to 4 ranges from least to most useful) among the design steps taught, and that the teaching course provided least guidance in this step (2.45 against an average of 2.72).

Table 1: Comparison of depth metric scores of students (31 out of 50) who used design the design process with those who did not (15 out of 50).

	Average score in depth metrics for students who used function structures approach		Average score in depth metrics for students who did not use the approach	
	For all 31 students	Without the 5 top scorers	For all 15 students	Without the 2 top scorers
Problem Broadening	-0.32258	- 0.53846	0	-0.0769231
Analysing existing solutions	-0.74194	-0.96154	-0.6666667	-0.8461538
Richness of solutions	-0.54839	-0.65385	-0.3333333	-0.5384615
Variety of solutions	-0.48387	-0.65385	-0.7333333	- 1
Total	-2.09677	-2.80769	-1.7333333	-2.4615385

Observations (Table 1, first two columns) indicate that the function structures approach did not foster deep understanding and problem solving. A score of 0, -1 or +1 respectively shows an average, bad or good performance. The second column shows how the average scores plummetted when the scores of the few students who did extremely well were excluded. The average score regarding any of the four metrics was negative, ie, below average. The understanding of the problem in the context of the existing solution had the worst score of all: important functions were often not used, and trivial ones used. Problems with the existing solution were rarely postulated, questioned or analysed. The given problem was rarely broadened beyond a single abstraction. The concepts generated were not rich in general (this has the second least score): functions existed without any apparent relation to each other (eg, park car, provide control, operate device), many functions failed to match levels (eg, *provide control* and *move vehicle* into position), functions were missing (eg, no stop function after move), and the process and function structures were not compatible. The solutions lacked variety: the use of function structure with morphological matrix did not help create a wide solution range, and although brainstorming gave a larger number, solutions lacked in richness.

Also observed was that it did not make a substantial difference as to whether or not students used function structures (compare columns 1 and 2 respectively with 2 and 4). Students who followed the design process did, on the whole, slightly worse that students who did not, but there were individual variations. Those who followed the approach had a larger variety but less richness of solutions, and an similar understanding of the existing solution in the context of the problem. The overall conclusion is that a lack of deep understanding and problem solving prevails but for few rare individuals (indicated by the substantial drop in average scores if the scores of these individuals are excluded), and remained a serious general problem.

4 Analysis: Inadequacies of the Existing Function Structure Approaches

4.1 Problems with Solution Neutral Elaboration of Function Structures

The above observations of students' lack of deep understanding and problem solving can be either because they do not understand the function structures approach well, or because the present understanding of function structures is poor.

We also noticed that, function structures in no circumstances were entirely solution-neutral. This may mean that students did not get expected results only because they did not identify the overall function, and develop function structures, in a solution-neutral way. So we looked into other literature to find any research where function structures were developed solution-neutrally and made a marked difference. In the case study by Ehrlenspiel and Dylla [1995] none of the three designers used function structures, even though two are scientific assistants in a school of engineering design! Schultheiss & Siers [1987] claimed they used generally valid functions for function structure development, but their work indicates that their function structure could not have been solution neutral, but possibly ensued from that of a human brick layer (there seems no other rationale for using *clamp* and *store* functions). Chakrabarti [1991] proposed a logical argument as to why function structures cannot be solution neutral while ensuring development towards solutions, and therefore a solution neutral detailing of function structures may not be useful. The earlier examples may be further evidences of this: if designers did not use function structures, it is because the method at present is not useful or easy to use, and if they did, they (implicitly) identified the functions from existing solutions. Recent evidence indicates that designers use tentative solutions to understand problems [Nidamarthi, 1996]; solution neutrality may not be desirable for problem understanding.

However, even this could be conjectural, so we sought if it was possible to have a single, solution-neutral function structure which was sufficiently close to a substantial proportion of the solutions generated by students so as to indicate their possibility. By abstracting the function structure sufficiently (eg, taking *move* and *space vehicles* as the two main functions) as suggested in Pahl and Beitz [1996], we could include many of the solutions. However, this abstraction only made the function structure too far removed from the solutions (such as bumpers on floor activated video-cameras) to indicate their existence to the unsuspecting designer. In other words, the abstraction encompassed the solutions, but not indicated them. The use of abstract I/O flows such as energy and signals only aggravates this problem.

Also, this abstract function structure fails to encompass many of the solutions considered. These solutions (eg, pressure balloon moved by car, or attached at the back of the vehicles, to keep vehicles spaced apart) requires thinking about the function structure in a solution-situated way (the partial solution of a physical stop to keep vehicles spaced apart needs to be conceived first, and only in the context of this, can the function of placing the stops between vehicles could be imagined). Another example is that a communication function would be required only when the agents doing various functions in a co-ordinated way (such as movement and control of the car) are already decided on, and are different from one another.

We argue that while the intention of keeping function structures solution-neutral is commendable, function structures are necessarily solution dependent. Even in the teaching notes in the course on conceptual design [Wallace, 1996], the examples given are solution dependent. In this lawn mower design example, the overall function is taken as *shorten grass*, which excludes the idea of having genetically engineered grass of constant height. Also, the function structure uses functions such as *cut grass* which presumes shortening to be done by cutting, and therefore excludes possibilities such as by spraying a chemical which eats away the extra length, or entices a bunch of rabbits to eat them, or developing grass which sheds off its extra length periodically, or grass which develops notches and needs just a tap to fall off. What we need is not necessarily to think in a solution neutral way, which is also unnatural, but be aware how and where the function structure is solution dependent, so that this information could be used to generate alternative function structures. One way to do this is to integrate function structure development process with the means which fulfil them.

4.2 Fostering Deep Understanding: The Information Required

The approach used earlier is too abstract and solution neutral to provide an effective understanding of the interplay of functions and solutions, ie, how a solution does or does not function. Central to deep understanding is understanding the solutions in the context of the problem. This needs elaborating their function structures to see what we know about the solutions and their problems, where the gaps in knowledge are, and if its partial solutions do not function as desired or are misused what their consequences might be.

Often the main sources of information from which an initial function structure can be drawn and analysed are the existing solutions and design problems envisaged in the design brief. However, often this brief is very sketchy, and expressed in natural languages. Natural language offers an enormous variety in what can be expressed; this includes precise as well as imprecise and incomplete knowledge. However, much of what is expressed is implicit or contextual. In order to support generation and elaboration of function structures which represent this, often informal, textual description of the problem, we require a systematic way of transforming such a description into a function structure. The advantage of this scheme would be to allow a rich visualisation of the existing solution in terms of its important functions, pre-requisites to their functioning, the inter-relationships between these functions, the means of fulfilling them, and the gaps in the resulting function structure description without which it is incomplete. We analyse below the design brief from the case study in order to identify the information a function structure must have. The brief is repeated below.

"If you wish to cross the English Channel by car through the Channel Tunnel, you must drive your vehicle into "Le Shuttle", the Channel Tunnel Train. Vehicles must enter the rear of the train from the side, as shown in the figure (not included here), and then drive towards the front until they approach the vehicle ahead. At this point, an official guides your vehicle forward, by using hand signals, until it is a short distance from the vehicle in front, when you are signalled to stop and put on the brake. This process is repeated for each vehicle in turn. All this is time consuming and requires an official for each train and deck."

We could identify the verbs which would probably be the *functions*. These are *cross* (the Channel), *drive* (vehicles into train), *enter* (the train), *drive* (towards the front), *approach* (vehicle ahead), *guide* (vehicle forward), *signal* (to stop and put on brake) *repeat* (for each vehicle) etc. As we see, often a number of *pre-conditions* (including input state and input flows) need to be satisfied before an operation or function can take place. For instance, vehicles must enter through the rear of the train from the side. Similarly there are some *post-conditions* until which a function should continue, and will have some *consequences* (such as an output state and flows), eg, the vehicle must drive towards the front until they approach the vehicle ahead. Moreover, often it is important to know *by whom* the function is achieved. For instance, drive vehicles into the train is achieved by the driver herself, or that signalling is done by an official. A related information is *how* each function is achieved. For instance guiding is done by the official using hand signals. There is often a causal or temporal *ordering* between these functions. For instance, *enter train* is before *drive towards front*. There can be a *"why is this function"* question, which is an inverse of *how*. For instance, the whole process is a prerequisite to *crossing* English Channel by "Le Shuttle". Information is often missing in this description, such as, how it is ensured that *drive towards front* achieves *approach vehicle ahead*. Information is often imprecise, such as what *approach the vehicle ahead* means: whether it the distance, and if so, how large. Many of the functions, such as *guide vehicles*, are implicit or simply vague, and need further clarification, requiring further detailing.

5 Discussion and Further Work

The are four major function structure representations, as described below.
Verb noun pair: the oldest, most versatile, but least amenable to deep computational support.
I/O representation: from Pahl and Beitz [1996] and many design researchers. Suitable for flow type functions (especially material flow, eg, harvest potatoes), but weak if no clear I/O flows exist (eg, support load). Functions are linked in a function structure via their I/O.
Algorithmic representation: Suitable for decision type functions, and used extensively in software designs. A function structure puts only logical ordering between its functions (eg, if the output of function A is this, then activate function B, otherwise activate function C).
State based representation: Here a function means a conversion between two states [Hubka, 1982], eg, support load defined as a load in a state of rest even of subjected to certain forces.

Although each of these approaches provide useful insight into the functionality of a system, none alone would be suffice to represent the whole complexity of a system's functionality, a glimpse of which is given in Section 4.2. I/O representations can represent the I/O aspects to provide clues to the additional functions required, and causal orderings between functions in a function structure. Algorithmic representations provide a way of ordering functions based on conditions. Verb-noun representations have the informality and flexibility that designers need. Missing in all these, however, is information about agents (who) and means (how), that could indicate how function structures can be abstracted or elaborated. A suitable approach must blend this information so as to allow the use of function structures for a deeper understanding and problem solving,. Further work involves developing and evaluating such an approach.

References:

Chakrabarti, A., "Designing by functions", PhD thesis, University of Cambridge, UK, 1991.
Ehrlenspiel, K. and Dylla, N. "Experimental Investigation of the Design Process", Proceedings ICED 1989, IME UK, 1989, pp 77-95.
Hubka, V., "Principles of Engineering Design, Butterworth Scientific", London,1988.
Nidamarthi, S. "The Co-Evolution of Requirements And Solutions in The Design Process", First Year Report, Cambridge University, Cambridge, UK, 1996.
Pahl, G., & Beitz, W., "Engineering Design: a systematic approach", Springer, London, 1996.
Roth, K., "Systematick der Machinen und ihrer Mechanischen Elementaren Funktionen", Feinwerktechnik, 74, 453-460, 1970.
Schultheiss, H. and Siers, F.J., "Experiences with Design Activities and Case Studies which are based on Methodical Design Processes", Proc. ICED 1987, Boston, 1987, pp 1074-1080.
Sturges, R.H., O'Shaughnessy, K., & Reed, R.G., "A systematic approach to conceptual design", Jl. of Concurrent Eng.: Research and Applications, Vol.1, No.2, 1993, pp 93-106.
Wallace, K., "An Introduction to the Engineering Design Process", Lecture Notes, Department of Engineering, University of Cambridge, UK, 1996.

Author's Name: Dr. Amaresh Chakrabarti
Institution/University: University of Cambridge
Department: Engineering Design Centre, Department of Engineering
Address: Trumpington Street, Cambridge CB2 1PZ
Country: United Kingdom
Telephone: ++ 01223 33 2828
Telefax: ++ 01223 33 2662
e-mail: ac123@eng.cam.ac.uk