The Function-Behaviour-Structure Ontology of Design

Chapter · February 2014

DOI: 10.1007/978-1-4471-6338-1_13

John Gero
GMU & UNCC

The Function-Behaviour-Structure Ontology of Design

John S Gero[1] and Udo Kannengiesser[2]

[1]Krasnow Institute for Advanced Study and University of North Carolina at Charlotte
[2]Metasonic AG, Pfaffenhofen, Germany

Abstract This chapter commences by introducing the background to the development of the Function-Behaviour-Structure (FBS) ontology. It then proceeds with an elaboration of the FBS ontology followed by the situated FBS framework which articulates a more detailed cognitive view. A series of exemplary empirical studies that use a coding scheme based on the FBS ontology is presented that demonstrates both the empirical support for the ontology and its applicability. The chapter concludes with a brief discussion on the role of this ontology and possible developments.

1. Introduction

Design is one of the profound activities of humans. It is the way humans intentionally change the physical and virtual worlds they inhabit. Society recognizes designing as important and privileges defined groups as designers such as engineers and architects, which are longstanding professions, along with relatively new groups such as software designers. It is therefore surprising that formal research into designing commenced relatively recently. Design research has largely adopted the scientific paradigm in which it is assumed that there are regularities that underlie phenomena and it is the role of research to discover and represent those regularities.

The early seminal works in design research in the 1960s and 1970s focused on methods and processes and produced an array of terminologies to describe designing. It was unclear whether the terms used by one group of researchers mapped onto terms used by other researchers or whether they were describing different phenomena. Design appeared to present problems for scientific research in that the results of the acts of designing were always unique and therefore there would be no regularity. This issue has been addressed in two ways. The first way was to look for underlying regularities in designs rather than surface features. The second way was to look for regularities in design processes. The term “designing” is used to signify the act and the term “design” is used to signify the result of designing.

This chapter presents the development of an approach to represent such regularities in designs and in designing. It commences with a brief introduction to the historical development of the concepts before expounding the Function-Behaviour-Structure (FBS) ontology of design and designing. This is followed by a section describing the situated Function-Behaviour-Structure (sFBS) ontology of design and designing. Empirical studies based on utilizing the FBS ontology are presented and provide experimental evidence in support of the ontology.

In searching for a way to think about designing, an axiom was proposed:

The foundations of designing are independent of the designer, their situation and what is being designed.

This has important consequences as it implies that the differences between design professions and design practices are not foundational to designing notwithstanding the apparent differences. The expectation was that the foundations of designing would not rely on any designing particulars.

Based on this axiom two hypotheses about representing designs and designing were proposed:

1. all designs could be represented in a uniform way, and
2. all designing could be represented in a uniform way.

What was being looked for was a set of irreducible foundational concepts of design and designing. These irreducible foundational concepts should cover the acts of designing and the representation of the design. Further, these irreducible foundational concepts should be distinct and have no overlap. In the 1980s a number of approaches to this were being developed by researchers that were based on the division of the design from the way it worked: Structure (S) for the design and Behaviour (B) for how it worked or performed. Many of these approaches used the term Function (F) to mean the intended behaviour of the design and as a consequence conflated Function and Behaviour and failed the no-overlap requirement.

Function-Behaviour-Structure was developed between 1984-86 and presented as part of a series of lectures on understanding design at Carnegie-Mellon University and at a seminar at Xerox PARC while the senior author was a consultant there in 1987. These presentations honed the understanding of the concepts. The ideas were presented at various conferences and resulted in the paper in a special issue on design in the *AI Magazine* in 1990 as part of a broader set of ideas (Gero 1990).

Clancey's 1997 book *Situated Cognition* (Clancey 1997) mapped well onto ill-formed concepts about the role the designer's cognitive understanding of the world inside their heads and around them as they designed. This led to the development of a cognitively richer articulation founded on FBS resulting in the situated Function-Behaviour-Structure (sFBS) framework of design and designing (Gero and Kannengiesser 2000, 2002, 2004).

Gruber developed the modern idea of an ontology (Gruber 1993). The notion of a foundational framework for the field of design mapped well onto the notion of an ontology since they both referred to the meta-level knowledge of a field. Thus, the FBS and sFBS frameworks became ontologies as frameworks for the knowledge in the field of designing.

Up to 1995 the FBS ontology was a conceptual construct that had been used to construct conceptual and computational models. Empirical studies of designing based on verbal protocol analysis (Ericsson and Simon 1993) had been introduced into design research some years earlier. These early studies and many of those continuing up to this day use project-specific schemes to code the protocol. The effect of this is that the results are incommensurable, ie, they cannot be compared to each other since the dimensions of what is being measured varies across projects. The FBS ontology offers a project-independent scheme to code the protocols. At the same time the ability of the FBS ontology-based coding scheme to capture the design-related utterances of designers in a protocol provides evidence of its utility if not its validity. This is not to claim that other coding schemes that take a different view of designing are not useful. The section on Empirical Studies demonstrates the wide-ranging applicability and utility the FBS ontology.

2. The Function-Behaviour-Structure Framework

The Function-Behaviour-Structure (FBS) ontology is a design ontology that describes all designed things, or artefacts, irrespective of the specific discipline of designing. Its three fundamental constructs – Function (F), Behaviour (B) and Structure (S) – are defined as follows:

Function is the teleology of the artefact ("what the artefact is for"). It is ascribed to the artefact by establishing a connection between one's goals and the artefact's measurable effects. Table 1 shows some examples of function of various artefacts.

Behaviour is defined as the artefact's attributes that can be derived from its structure ("what the artefact does"). Behaviour provides measurable performance criteria for comparing different artefacts. The examples of behaviour in Table 1 show that most instances of behaviour relate to notions of quality, time and cost.

Structure is defined as its components and their relationships ("what the artefact consists of"). The various examples of structure in Table 1 indicate that this definition can cover any physical, virtual or social artefact.

Humans construct connections between function, behaviour and structure through experience and through the development of causal models based on interactions with the artefact. Specifically, function is ascribed to behaviour by establishing a teleological connection between the human's goals and the observable or measurable performance of the artefact. Behaviour is causally connected to structure, i.e. it can be derived from structure using physical laws or heuristics. There is no direct connection between function and structure (De Kleer and Brown 1984).

Table 1. Examples of function, behaviour and structure of different artefacts

	Dwelling	Editing software	Manufacturing process	Team
Function (F)	Provide safety, provide comfort, provide affordability	Be time efficient, provide affordability	Be safe, be time efficient, provide sustainability, provide affordability	Be time efficient, provide affordability
Behaviour (B)	Strength, weight, heat absorption, cost	Response times, cost	Throughput, accuracy, speed, waste rate, cost	Working speed, success rate, cost
Structure (S)	Geometrically interconnected walls, floors, roof, windows, doors, pipes, electrical systems	Computationally interconnected program components	Logically and physically interconnected operations and flows of material and information	Socially interconnected individuals

The FBS framework (Gero 1990) is an extension of the FBS ontology to represent the process of designing as a set of transformations between function, behaviour and structure. The most basic view of designing consists of transformations from function to behaviour, and from behaviour to structure:

(1) F → B, and

(2) B → S

In this view, behaviour is interpreted as the performance expected to achieve desired function. Yet, once a structure is produced, it must be checked whether the artefact's "actual" performance, based on the structure produced and the operating environment, matches the "expected" behaviour. Therefore, the FBS framework distinguishes two classes of behaviour: expected behaviour (Be) and behaviour derived from structure (Bs). This extends the set of transformations with which we can describe designing to include:

(1) F → Be,

(2) Be → S,

(3) S → Bs, and

(4) Be ↔ Bs (comparison of the two types of behaviour).

The observable input and output of any design activity is a set of requirements (R) that come from outside the designer and a description (D) of the artefact, respectively. The FBS framework subsumes R in the notion of function and defines D as the external representation of a design solution:

(5) S → D

Based on the common observation that designing is not only a process of iterative, incremental development but frequently involves focus shifts, lateral thinking and emergent ideas, the FBS framework defines the following additional transformations:

(6) S → S',

(7) S → Be', and
(8) S → F (via Be).

These three transformations assume an existing structure as the driver for generating changes in structure, behaviour or function.

The eight fundamental transformations or processes in the FBS framework are shown and labelled in Figure 1:

1. Formulation (R → F, and F → Be)
2. Synthesis (Be → S)
3. Analysis (S → Bs)
4. Evaluation (Be ↔ Bs)
5. Documentation (S → D)
6. Reformulation type 1 (S → S')
7. Reformulation type 2 (S → Be)
8. Reformulation type 3 (S → F (via Be))

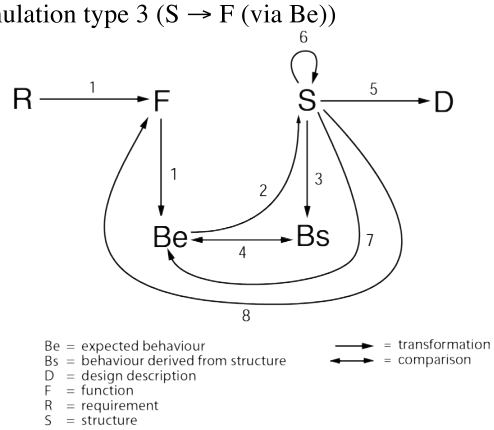

Fig. 1. The FBS framework

The FBS framework represents the beginnings of a theory of designing, through its ability to describe any instance of designing irrespectively of the specific domain of design or the specific methods used. Section 4 will present how empirical studies provide a validation of the FBS framework in the sense of a theory of designing.

3. The Situated Function-Behaviour-Structure Framework

The situated FBS framework was developed in 2000 as an extension of the FBS framework to include the notion of situatedness (Gero and Kannengiesser 2000). It is founded on the idea that situated designing involves interactions between three worlds: the external world, the interpreted world and the expected world, Figure 2.

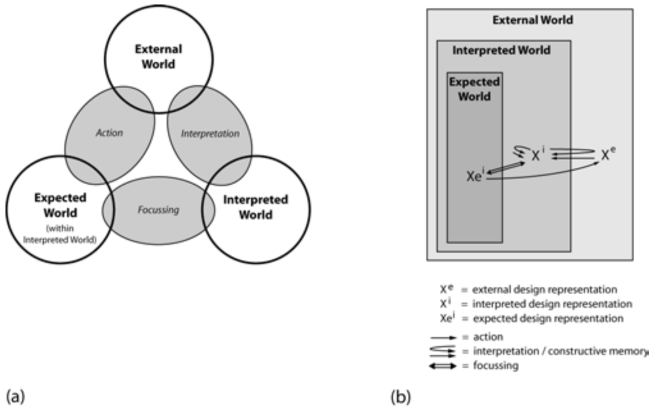

Fig. 2. Situatedness as the interaction of three worlds: (a) general model, (b) specialised model for design representations

The *external world* is the world that is composed of things outside the designer. No matter whether things are "real" or represented, we refer to all of them as just "design representations". This is because their purpose is to support interpretation and communication of designers.

The *interpreted world* is the world that is built up inside the designer in terms of sensory experiences, percepts and concepts. It is the internal representation of that part of the external world that the designer interacts with. The interpreted world provides an environment for analytic activities and discovery during designing.

The *expected world* is the world imagined actions of the designer will produce. It is the environment in which the effects of actions are predicted according to current goals and interpretations of the current state of the world.

These three worlds are related together by three classes of interaction. *Interpretation* transforms variables that are sensed in the external world into sensory experiences, percepts and concepts that compose the interpreted world. *Focussing* takes some aspects of the interpreted world and uses them as goals for the expected world. *Action* is an effect which brings about a change in the external world according to the goals in the expected world.

Figure 2(b) presents a specialised form of this model, with the designer (described by the interpreted and expected world) located within the external world, and with general classes of design representations placed into this nested model. The set of expected design representations (Xe^i) corresponds to the notion of a design state space, i.e. the state space of all possible designs that satisfy the set of requirements. This state space can be modified during the process of designing by transferring new interpreted design representations (X^i) into the expected world and/or transferring some of the expected design representations (Xe^i) out of the expected world. This leads to changes in external design representations (X^e), which may then be used as a basis for re-interpretation changing the interpreted world. Novel interpreted design representations (X^i) may also be the result of memory (here called *constructive memory*), which can be viewed as a process of interaction among design representations within the interpreted world rather than across the interpreted and the external world.

Both interpretation and constructive memory are viewed as "push-pull" processes, i.e. the results of these processes are driven both by the original experience ("push") and by some of the agent's current interpretations and expectations ("pull") (Gero and Fujii 2000). This notion captures two ideas. First, interpretation and constructive memory have a subjective nature, using first-person knowledge grounded in the designer's interactions with their environment (Bickhard and Campbell 1996; Clancey 1997; Ziemke 1999; Smith and Gero 2005). This is in contrast to static approaches that attempt to encode all relevant design knowledge prior to its use. Anecdotal evidence in support of first-person knowledge is provided by the common observation that different designers perceive the same set of requirements differently (and thus produce different designs). And the same designer is likely to produce different designs at later times for the same requirements. This is a result of the designer acquiring new knowledge while interacting with their environment between the two times.

Second, the interplay between "push" and "pull" has the potential to produce emergent effects, leading to novel and often surprising interpretations of the same internal or external representation. This idea extends the notion of biases that simply reproduce the agent's current expectations. Examples have been provided from experimental studies of designers interacting with their sketches of the design object. Schön and Wiggins (1992) found that designers use their sketches not only as an external memory, but also as a means to reinterpret what they have drawn, thus leading the design in a surprising, new direction. Suwa et al. (1999) noted, in studying designers, a correlation of unexpected discoveries in sketches with the invention of new issues or requirements during the design process. They concluded that "sketches serve as a physical setting in which design thoughts are constructed on the fly in a situated way". Guindon's (1990) protocol analyses of software engineers, designing control software for a lift, revealed that designing is characterized by frequent discoveries of new requirements interleaved with the development of new partial design solutions. As Guindon puts it, "designers try to make the most effective use of newly inferred requirements, or the sudden discovery of partial solutions, and modify their goals and plans accordingly".

Gero and Kannengiesser (2000, 2002, 2004) have combined the FBS framework with the model of interacting worlds, by specialising the model of situatedness shown in Figure 2(b). In particular, the variable X, which stands for design representations in general, is replaced with the more specific representations F, B and S. This provides the basis of the situated FBS framework, Figure 3. In addition to using external, interpreted and expected F, B and S, this framework uses explicit representations of external requirements, represented as external requirements on function (FR^e), external requirements on behaviour (BR^e), and external requirements on structure (SR^e). The situated FBS framework also introduces the process of comparison between interpreted behaviour (B^i) and expected behaviour (Be^i), and a number of processes that transform interpreted structure (S^i) into interpreted behaviour (B^i), interpreted behaviour (B^i) into interpreted function (F^i), expected function (Fe^i) into expected behaviour (Be^i), and expected behaviour (Be^i) into expected structure (Se^i). Figure 3 uses the numerals 1 to 20 to label the resultant set of processes; however, they do not represent any order of execution.

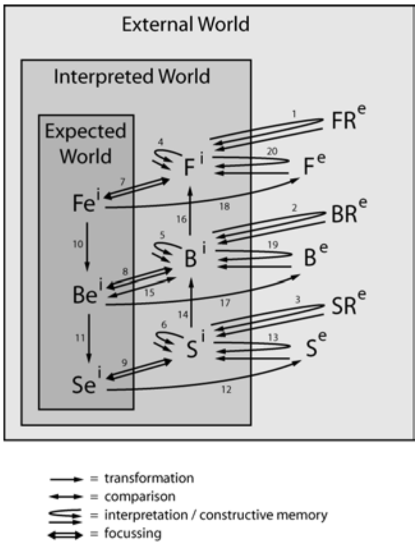

Fig. 3. The situated FBS framework (Gero and Kannengiesser 2004)

The 20 processes in the situated FBS framework map onto the eight fundamental processes in the original FBS framework. The remainder of Section 3 presents these mappings, and illustrates them using a turbocharger as the artefact.

3.1 Formulation

Formulation frames the design task by defining a state space of potential design solutions (structure state space) and a set of criteria for assessing these solutions (behaviour state space). This activity uses a set of goals (function state space) and constraints that are given to the designer by external specification or are constructed based on the designer's own experience. In the situated FBS framework, formulation includes processes 1 to 10, Figure 4.

Example: A turbocharger designer is provided with a set of requirements by an automobile company that include:

- FR^e: increase the power output of a specific engine of a specific passenger car
- BR^e: air mass flow and efficiency ratio for a range of different engine speeds

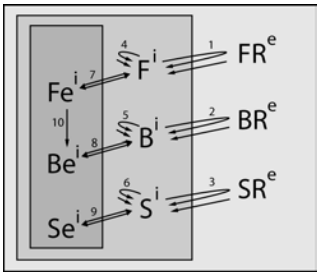

Fig. 4. Formulation

- SR^e: maximal spatial dimensions; position of connecting points to other components

These requirements are interpreted to produce F^i, B^i and S^i (processes 1, 2 and 3) that are complemented with implicit requirements constructed from the designer's memory (processes 4, 5 and 6). These additional requirements include:

- F^i: provide reliability, provide reduced manufacturing cost
- B^i: ranges of values for the pressure ratio of compressor and turbine at the different engine speeds
- S^i: basic components (compressor, turbine, core assembly) and their parameters including geometrical variables and classes of material (e.g., aluminum for compressor, and cast iron for turbine); ranges of values for inlet and outlet diameters of compressor and turbine

Processes 7, 8 and 9 represent deciding on a set of turbocharger requirements to form the design state space. Process 10 captures how additional expected behaviour (Be^i) is derived from expected function (Fe^i). For example, expected ranges of thermal strength are derived from the function requirement of reliability.

3.2 Synthesis

Synthesis instantiates a design solution in terms of a point in the structure state space. It includes processes 11 and 12, Figure 5.

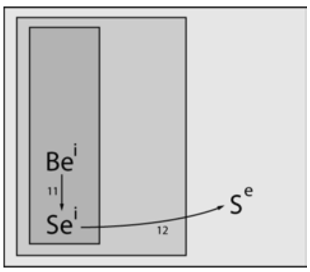

Fig. 5. Synthesis

Example: The designer produces a design by deciding on the values of the formulated structure variables for the turbocharger (process 11). The design is then externalized (process 12) as a drawing on paper, as a computational model using a computer-aided drafting (CAD) tool, or as a physical prototype.

3.3 Analysis

Analysis derives the behaviour from the design solution. It includes processes 13 and 14, Figure 6.

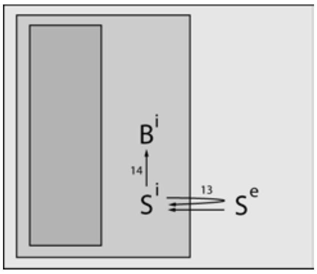

Fig. 6. Analysis

Example: The designer uses a range of calculations, simulations and physical prototype tests to analyse the design solution of the turbocharger. This requires interpretation of external structure (process 13), either by the designer or an engineer testing a prototype or visually inspecting iconic or mathematical models, or by an analysis tool that reads CAD files. Behaviour can then be derived (process 14) in one of three ways:

- By computation: Specialized tools are used to perform complex calculations and simulations. For example, thermal strength of turbochargers (particularly of their turbine components) is often derived using a finite-element analysis (FEA) tool.
- By physical measurement: Behaviours can be derived from the physical, electrical or chemical effects caused by the interaction of measurement devices and physical prototypes. This is frequently used in turbocharger analysis, to derive pressures and temperatures produced by turbines and compressors under realistic operating conditions.
- By human reasoning: This is done only for very simple derivations of behaviour and usually involves extensive use of external memory aids. Human reasoning is best applied in combination with computation or physical measurement. For example, dividing the compressor's inflow pressure (an exogenous variable) by its outflow pressure (a behaviour measured in a prototype test) is a

simple calculation that produces the compressor's pressure ratio (a derivative behaviour).

3.4 Evaluation

Evaluation assesses the design solution on the basis of the formulated criteria, i.e. by comparison of the behaviour derived from the design solution and the expected behaviour. Evaluation includes process 15, Figure 7.

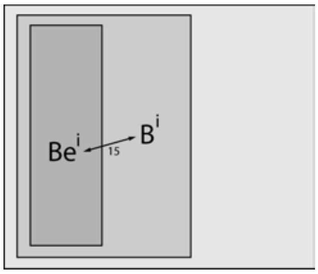

Fig. 7. Evaluation

Example: The designer compares the air mass flows, pressure ratios, strength etc. analysed with the ones required (process 15). Based on the outcome of this comparison, the designer decides whether the design of the turbocharger satisfies the requirements. In most cases, changes are needed that lead to further cycles of synthesis, analysis and evaluation. For example, the turbine's pressure ratio may be evaluated as too low to achieve required mass flow rates. The designer may then decide to synthesize a modified structure with larger values for the turbine wheel's geometric variables.

3.5 Documentation

Documentation produces an external representation of a design solution for purposes of communicating that solution. In most instances of designing "physical" products, this step is required to provide the builder or manufacturer with a "blueprint" for realizing the product. Documentation includes processes 12, 17 and 18, Figure 8.

Example: After successful evaluation of the turbocharger, a number of drawings and CAD models are produced of the assembly including its individual components (process 12) so that the turbocharger can be manufactured. A number of diagrams documenting some of the behaviour, such as efficiency, air mass flow and pressure ratio, are also generated (process 17) as "performance maps" for the automobile company. Some functions may be documented for purposes of indexing, marketing or explaining design decisions (process 18).

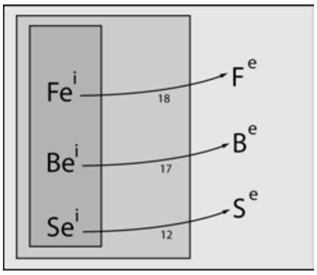

Fig. 8. Documentation

3.6 Reformulation Type 1

Reformulation type 1 reframes the structure state space, directly creating a new space of possible designs. This often entails a subsequent modification of the behaviour state space. Reformulation type 1 includes process 9, Figure 9. Processes 3, 6 and 13 are the potential drivers of this type of reformulation, as they all have the potential to produce new structure.

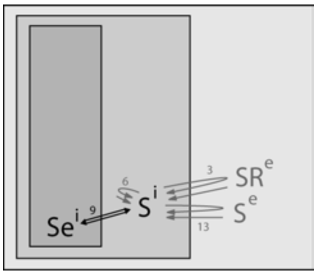

Fig. 9. Reformulation type 1

Example: The designer may decide to extend the ranges of values of the turbocharger's geometric dimensions (process 9), such that it allows the selection of much smaller values than previously expected. This can be seen as creating a new family of (smaller) turbocharger variants. The decision to reformulate structure may be the result of external drivers, such as new external requirements from the car manufacturer (process 3) or studies of a competitor's product (process 13), or an internal driver, such as reflection on integrating new technologies (e.g., new materials) (process 6).

3.7 Reformulation Type 2

Reformulation type 2 reframes the behaviour state space. In most cases, this leads to a modification of the structure state space, and thus to the creation of a new space of possible designs. In some cases, the new behaviour may also drive changes in the set of functions. Reformulation type 2 includes process 8, Figure 10. Processes 2, 5, 14 and 19 are the potential drivers of this type of reformulation, as they all have the potential to produce new behaviour.

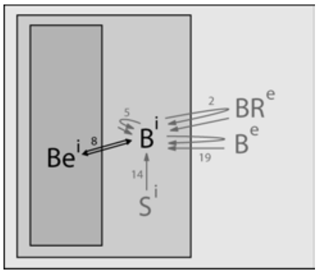

Fig. 10. Reformulation type 2

Example: The designer may want to introduce a new control behaviour for varying the air mass flow. This leads to the creation of a design state space with new characteristics that become visible through changes in structure. Possible changes of the turbocharger's structure are the addition of variable guide vanes or a variable sliding ring inside the turbine. The reformulation of the turbocharger's control behaviour may be the result of external drivers, such as new external requirements from the car manufacturer (process 2) or the interpretation of ideas articulated in a brainstorming meeting (process 19), or internal drivers, such as reflection on previous experiences regarding variable control (process 5) or analogical derivation of behaviour from structurally related objects (e.g., water turbines with variable inlet nozzle sizes to control water supply) (process 14).

3.8 Reformulation Type 3

Reformulation type 3 reframes the function state space. In most cases, this leads to a modification of the behaviour and structure state space, and thus to the creation of a new space of possible designs. Reformulation type 3 includes process 7, Figure 11. Processes 1, 4, 16 and 20 are the potential drivers of this type of reformulation, as they all have the potential to produce new function.

Example: Supporting modified engine characteristics represents new function requirements for the turbocharger. For example, turbocharging an engine with significantly increased exhaust temperature may affect the thermal strength such that a more resistant class of material needs to be chosen for the turbine. The reformulation of the turbocharger's function may be the result of external drivers, such as new external requirements from the car manufacturer (process 1) or the interpretation of alternative functions expressed in a morphological matrix (process 20), or internal drivers, such as reflection on previous experiences with products of high temperature resistance (process 4) or analysis of potential consequences of technological improvements regarding thermal strength (process 16).

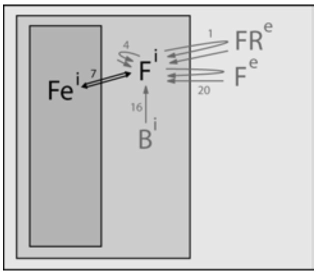

Fig. 11. Reformulation type 3

The situated FBS framework represents a further development towards a theory of designing, by accounting for the dynamics of the situation within which most instances of designing occur. Section 4 will present how empirical studies provide a validation of the situated FBS framework in the sense of a theory of designing.

4. Empirical Studies

Verbal protocol analysis is a rigorous methodology for eliciting verbal reports of thought sequences as a valid source of data on thinking. It is a well-developed, validated method for the acquisition of empirical data on thinking (Crutcher 1994; Ericsson and Simon 1993; Van-Someren et al 1994). The generic process of protocol analysis results in a sequence of codes that represent the cognitive activations during thinking. Using the FBS ontology as the basis of the coding scheme produces results that are commensurable across protocols independent of the designer, the design task, and all aspects of the design environment. The FBS codes represent the cognitive activations of the design issues that the designers are thinking about as they are designing. The FBS-based design processes, that are a consequence of the transformations of the design issues, Figure 1, are available from the coding of the protocols (Kan and Gero 2009).

Such empirical studies can be used to test the utility of the FBS ontology by measuring the percentage of design-related utterances not covered by the FBS coding as well as being used to characterize the cognition of designing. In a wide range of protocol studies the percentage of design-related utterances not covered in any protocol has been zero or diminishingly small. This does not imply that the FBS ontology-based coding is the only coding scheme that covers empirical data about designing, rather the implication is that the FBS ontology provides a robust foundation for the development of a generic coding scheme. Protocols coded using the FBS coding are commensurable. Results from FBS coded protocols provide insight into designing and confirm the utility of the FBS ontology. A small number of such results is presented below to provide indicate exemplars of what can be found using this approach.

4.1 Comparing Different Disciplines Designing

The question of what are the differences between different disciplines as they are designing can be addressed through empirical studies. The results of a set of studies of architects, software designers and mechanical engineers designing in terms of their respective design issue distributions are shown in Figure 12. The use of the FBS coding scheme allows for a direct comparison. These results from these studies indicate that architects spend more of their cognitive effort on the design issue of function than do software designers and mechanical engineers. Mechanical engineers spend more of their cognitive effort on behavior from structure and less on expected behavior than do architects and software designers.

These results provide evidentiary support for the claim that the FBS ontology can be used independently of design discipline and design task.

4.2 Comparing High School and University Students Designing

Do high school and university students design differently? An experiment was conducted where high school students and sophomore (second year) mechanical engineering university students were given the same design task. The results of their design issue distributions are presented in Figure 13.

These results show that university students have a different distribution of their cognitive effort than do high school students. That difference manifests itself primarily in the differences in both expected behavior and behavior from structure.

4.3 Comparing Effects of Using Different Design Techniques

Does teaching different concept generation techniques result in different design behaviors? The same cohort of students was taught brainstorming, morphological analysis and TRIZ. After learning each technique the cohort carried out a design task using the technique just learned. The results of measuring the distributions of design issues when utilizing each concept generation technique are presented in Figure 14.

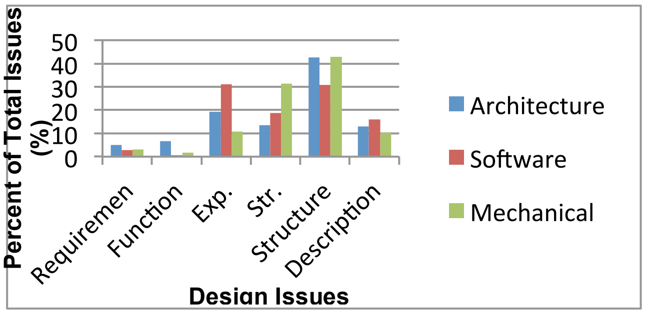

Fig. 12. Comparing the design issue distributions of three different design disciplines (after Kan and Gero 2011a).

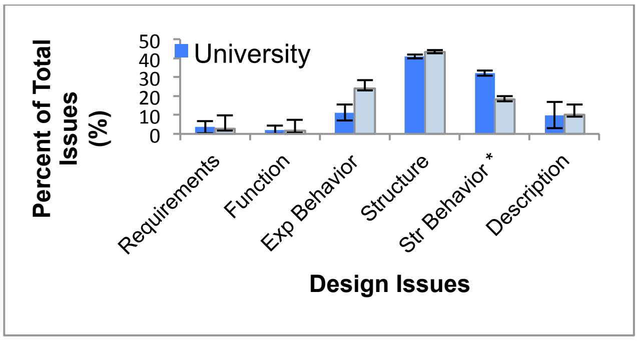

Fig. 13. Comparing the design issue distributions of high school students and university students (after Lammi and Gero 2011).

These results indicate that the use of different concept generation techniques produces different cognitive behavior in the designers. The most significant difference manifests itself in the increased cognitive effort expended on function and expected behavior when using TRIZ compared to the other two methods.

4.4 Who is Doing What in a Design Team?

As designing is increasingly carried in teams, the behavior of design teams and the individuals in them become of interest. One characterization that provides access to the behavior of teams and individuals in teams is the design process. In the FBS ontology design processes are the transformations from one design to another.

This is represented by the semantic linkograph of the protocol (Goldschmidt 1990, Kan and Gero 2009). The linkograph of a team of designers provides the basis to extract the design process of individuals and to articulate which members of the team are involved in each process.

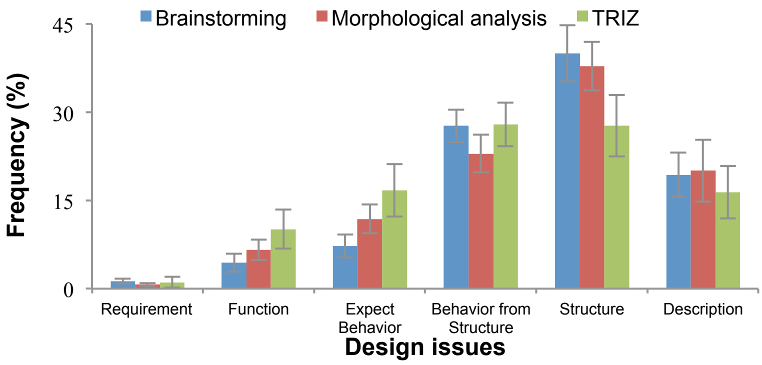

Fig. 14. Comparing the design issue distributions when designing with three different concept generation techniques. Data is from undergraduate engineering students (after Gero et al 2012).

The design issue at the end of each link generates the design process and linking the names of the individuals associated with each end of a design process provides a highly detailed description of the design process involvement of each individual during a design session.

In a study of a design team in industry the synthesis process of Allan, the team leader, is extracted from the protocol's semantic linkograph in such a way that his interactions with each member of the 7-person team can be followed, Figure 15.

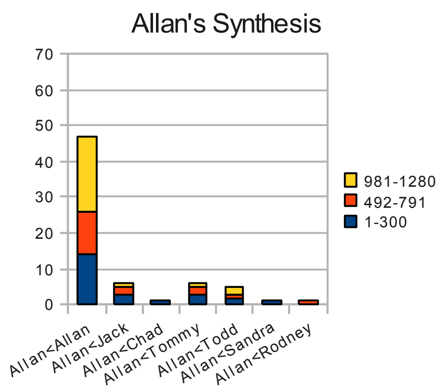

Fig. 15. Allan's (the team leader) synthesis process (Be -> S) in terms of interactions with other team members, presented as a stacked histogram of the results derived from dividing the design session into three thirds (defined by their segment number ranges on the right of the graph). This allows for comparing the time behaviour of synthesis interactions across the design session. Data is from a design team in industry (after Kan and Gero 2011b).

These results demonstrate the wide-ranging applications of utilizing a coding scheme based on the FBS ontology.

5. Discussion

The validation of methods, models and theories in design is a process of building confidence in their usefulness (Pedersen et al. 2000). An increasing number of studies are supporting such confidence for the FBS ontology of design. They provide evidence for its applicability, and for the tools it can offer for understanding designing and designs.

The applicability of the FBS ontology is shown through its large coverage that has been demonstrated conceptually and empirically. Conceptually, the FBS ontology has been used in various design domains including architectural, mechanical, software, and business process design (Bergmann 2002, Erdman 2008, Kruchten 2005, Wilke 1999) to represent designs and design processes as a basis for methodologies and computational models (Liew and Gero 2004; Kannengiesser and Gero 2006). Empirically, the FBS ontology has been used for coding hundreds of design protocols representing design processes that varied along multiple dimensions such as the designers' expertise and discipline, the design task, and the size and composition of the design team.

The FBS ontology provides a number of tools for understanding designing and designs. The FBS and sFBS frameworks, in particular their graphical depictions, are tools for understanding designing in terms of its fundamental processes and its situatedness, respectively. They have been used for understanding a process not directly connected to designing: how people construct affordances of a designed object (Kannengiesser and Gero 2012). The FBS-based annotations proposed by Kannengiesser (2010) are a tool for understanding a more abstract class of designs, business process designs. Some well-developed analysis tools for design protocols are also based on the FBS ontology (Gero et al 2011); they include the entropy of semantic linkographs for measuring the creation of novel concepts during designing (Kan and Gero 2008) and the problem-solution (P-S) index for measuring the relative cognitive effort spent on either the problem or the solution (Jiang et al 2013).

A limitation of the high level of generality of the FBS ontology is that some specific aspects of designing are not directly addressed. Articulating the FBS ontology to map onto other framework descriptions of design may demonstrate more detailed areas of coverage that to date are not immediately obvious. For example, subclasses of function, behaviour and structure may be defined to represent different levels in a compositional hierarchy. Transformations between subclasses of the same ontological class but at different hierarchical levels would then represent processes of composition or decomposition, which are commonly described activities in other models of designing (Sim and Duffy 2003).

The FBS ontology has been shown to be a robust descriptor of designs and designing. The ontology continues to be widely cited with an average of two to three citations a week for the last decade.

Acknowledgments

The research reported here has been supported by multiple grants from the Australian Research Council, the US National Science Foundation (NSF Grant Nos: EEC-0934824, CMMI-0926908, IIS-1002079, CMMI-1161715) and the US Defense Advanced Projects Agency (DARPA). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Bergmann, R: 2002, *Experience Management: Foundations, Development Methodology, and Internet-based Applications*, Springer.

Bickhard, MH and Campbell, RL: 1996, Topologies of learning, *New Ideas in Psychology* **14**(2): 111-156.

Clancey, WJ: 1997, *Situated Cognition: On Human Knowledge and Computer Representations*, Cambridge University Press, Cambridge.

De Kleer, J and Brown, JS: 1984, A qualitative physics based on confluences, *Artificial Intelligence* **24**: 7-83.

Erden, MS et al: 2008, A review of function modeling: Approaches and applications, *AIEDAM* **22**(2): 147-169.

Gero, JS: 1990, Design prototypes: A knowledge representation schema for design, *AI Magazine* **11**(4): 26-36.

Gero, JS and Fujii, H: 2000, A computational framework for concept formation for a situated design agent, *Knowledge-Based Systems* **13**(6): 361-368.

Gero, JS, Kan, JWT and Pourmohamadi, M: 2011, Analysing design protocols: Development of methods and tools, *in* A Chakrabarti (ed.), *Research into Desig*n, Research Publishing, pp. 3-10.

Gero, JS and Kannengiesser, U: 2000, Towards a situated function-behaviour-structure framework as the basis of a theory of designing, *in* T Smithers (ed.) *Workshop on Development and Application of Design Theories in AI in Design Research*, Artificial Intelligence in Design'00, Worcester, MA, pp. gk:1-5.

Gero, JS and Kannengiesser, U: 2002, The situated function-behaviour-structure framework, *in* JS Gero (ed.) *Artificial Intelligence in Design'02*, Kluwer, Dordrecht, pp. 89-104.

Gero, JS and Kannengiesser, U: 2004, The situated function-behaviour-structure framework, *Design Studies* **25**(4): 373-391.

Gero, JS, Jiang, H and Williams, CB (2012) Does using different concept generation techniques change the design cognition of design students? *ASME IDETC*, DETC2012-71165

Guindon, R: 1990, Designing the design process: Exploiting opportunistic thoughts, *Human-Computer Interaction* **5**: 305-344.

Jiang, H, Gero, JS and Yen, CC: 2013, Exploring designing styles using Problem-Solution indexes, *Design Computing and Cognition'12*, Springer (to appear)

Kan, J and Gero, JS: 2008, Acquiring information from linkography in protocol studies of designers, *Design Studies* **29**(4): 315-337.

Kan, JWT and Gero JS: 2009, Using the FBS ontology to capture semantic design information in design protocol studies, *in* J McDonnell and P Lloyd (eds) *About: Designing. Analysing Design Meetings*, CRC Press, pp. 213-229.
Kan, JWT and Gero, JS: 2011a, Comparing designing across different domains: An exploratory case study, *in* S Culley, B Hicks, T McAloone, T Howard and Y Reich (eds) *Design Theory and Methodology,* Design Society, Glasgow, pp. 2:194-203.
Kan, JWT and Gero, JS: 2011b, Learning to collaborate during team designing: Quantitative measurements, *in* A Chakrabarti (ed.) *Research into Design*, Research Publishing, pp. 687-694.
Kannengiesser, U: 2010, Towards a methodology for flexible process specification, *Entreprise Modelling and Information Systems Architectures* **5**(3): 44-63.
Kannengiesser, U and Gero, JS: 2006, Towards mass customized interoperability, *Computer-Aided Design* **38**(8): 920-936.
Kannengiesser, U and Gero, JS: 2012, A process framework of affordances in design, *Design Issues* **28**(1): 50-62.
Kruchten, P: 2005, Casting software design in the function-behavior-structure framework, *IEEE Software* **22**(2): 52-58.
Lammi, M and Gero, JS: 2011, Comparing design cognition of undergraduate engineering students and high school pre-engineering students*, 41st ASEE/IEEE Frontiers in Education*, ASEE/IEEE, pp. F4F:1-6.
Liew, P-S and Gero, JS: 2004, Constructive memory for situated design agents, *AIEDAM* **18**(2): 163-198.
Pedersen, K, Emblemsvag, J, Bailey, R, Allen JK and Mistree, F: 2000, Validating design methods & research: The validation square, *Proceedings of DETC '00, 2000 ASME Design Engineering Technical Conferences*, Baltimore, MD, paper no. DETC2000DTM-14579.
Schön, DA and Wiggins, G: 1992, Kinds of seeing and their functions in designing, *Design Studies* **13**(2): 135-156.
Sim, SK and Duffy, AHB: 2003, Towards an ontology of generic engineering design activities, *Research in Engineering Design* **14**(4): 200-223.
Smith, GJ and Gero, JS: 2005, What does an artificial design agent mean by being 'situated'?, *Design Studies* **26**(5): 535-561.
Suwa, M, Gero, JS and Purcell, T: 1999, Unexpected discoveries and s-inventions of design requirements: A key to creative designs, *in* JS Gero and ML Maher (eds) *Computational Models of Creative Design IV,* Key Centre of Design Computing and Cognition, University of Sydney, Sydney, Australia, pp. 297-320.
Wilke, W: 1999, Knowledge Management for Intelligent Sales Support in Electronic Commerce, *Doctoral Dissertation*, University of Kaiserslautern, Germany.
Williams, CB, Gero, JS, Lee, Y and Paretti, M: 2011, Exploring the effect of design education on the design cognition of mechanical engineering students, *ASME IDETC201,* 1DETC2011-48357.
Ziemke, T: 1999, Rethinking grounding, *in* A Riegler, M Peschl and A von Stein (eds) *Understanding Representation in the Cognitive Sciences: Does Representation Need Reality?*, Plenum Press, New York, pp. 177-190.

To appear as: Gero, JS and Kannengiesser, U (2013) The Function-Behaviour-Structure ontology of design, *in* Amaresh Chakrabarti and Lucienne Blessing (eds), *An Anthology of Theories and Models of Design*, Springer.