An Ontological Basis for Design Methods

KANNENGIESSER, Udo

An Ontological Basis for Design Methods

Udo Kannengiesser, NICTA, Australia, and School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

Abstract

This paper presents a view of design methods as process artefacts that can be represented using the function-behaviour-structure (FBS) ontology. This view allows identifying five fundamental approaches to methods: black-box, procedural, artefact-centric, formal and managerial approaches. They all describe method structure but emphasise different aspects of it. Capturing these differences addresses common terminological confusions relating to methods. The paper provides an overview of the use of the fundamental method approaches for different purposes in designing. In addition, the FBS ontology is used for developing a notion of prescriptiveness of design methods as an aggregate construct defined along four dimensions: certainty, granularity, flexibility and authority. The work presented in this paper provides an ontological basis for describing, understanding and managing design methods throughout their life cycle.

Keywords

Design Methods; Function-Behaviour-Structure (FBS) Ontology; Prescriptive Design Knowledge

Design methodology is an area of research that is concerned with the development, application and validation of design methods. Work on design methods has been carried out in a number of design disciplines, particularly in engineering design. These methods aim to guide designers (or design systems) solving recurrent classes of design problems, thus enhancing the quality of design outcomes and the efficiency of design processes. Methods are crucial not only for educating novice designers, but also for managing the activities of expert designers according to the goals and constraints of particular design projects.

The nature, scope and research approaches of design methodology have been well described (Eekels and Roozenburg 1991; Pedersen et al. 2000; Frey and Dym 2006). However, most descriptions of design methods that are the subject of this field convey a rather vague understanding of some of its fundamental concepts. Specifically, two aspects of design methods have not been well addressed:

- *Terminology/Typology*: Some design researchers use the term "method" interchangeably with a wide array of terms, such as "notation", "model", "process", "technique" and "tool". Others seem to distinguish between some these terms, but without articulating what it is that differentiates them. This leads to conceptual ambiguities and miscommunication among design scholars. What is needed is a

general framework for design methods that makes explicit the differences and interrelationships between various method aspects.

- *Prescriptiveness*: It is generally accepted that design methods represent prescriptive rather than descriptive knowledge about designing (Vermaas and Dorst 2007). On the other hand, designers need to have sufficient "realisation freedom" (van Aken 2005) to adapt the application of a design method to the situation at hand. It is necessary to be explicit about which parts of a method provide binding constraints for the designer's actions and to what extent. This requires a definition of prescriptiveness that is more differentiated than its common interpretation as a "to-be" (as opposed to an "as-is") state of affairs.

This paper addresses these issues by proposing an ontological basis for characterising design methods in a uniform way, independently of the particular domain of designing and the specific terms used. This enables a better understanding of methods both across and within design disciplines, which may lead to improved modelling and management of design methods. The function-behaviour-structure (FBS) ontology (Gero and Kannengiesser 2004) provides the foundations for this study. Although most examples presented in the paper are predominantly from engineering design, we posit that the underpinning ideas are applicable to any other design discipline.

The Function-Behaviour-Structure Ontology

The FBS ontology distinguishes between three aspects of an artefact: function (F), behaviour (B) and structure (S). This ontology has been applied to objects (Gero 1990; Gero and Kannengiesser 2004) and processes (Gero and Kannengiesser 2007).

- *Function* (F) of an artefact is defined as its teleology, i.e. "what the artefact is for". An example is the function "to wake someone up" that humans generally ascribe to the behaviour of an alarm clock. The notion of function is independent of whether the artefact is an object or a process.
- *Behaviour* (B) of an artefact is defined as the attributes that are derived or expected to be derived from its structure, i.e. "what the artefact does". An example of object behaviour is "weight", which can be derived directly from a physical object's structure properties of material and spatial dimensions. Typical behaviours of processes include speed, cost, precision and accuracy.
- *Structure* (S) of an artefact is defined as its components and their relationships, i.e. "what the artefact consists of". It represents the artefact's "building blocks" that can be directly created or modified by the designer. Structure can be classified as macro-structure or micro-structure. Macro-structure comprises the set of components and relationships that are distinguishable at a given level of abstraction. For physical objects, this includes their geometry. For processes, this includes their input (i), transformation (t) and output (o). Micro-structure comprises those components and relationships that are too fine-grained to be represented explicitly, and is only described using a

shorthand label. For physical objects, this includes their material. For processes, this includes the agent performing the transformation, where the "agent" can be viewed in an object-centred way (e.g., as a person or a software system) or in a process-centred way (i.e., as a mechanism composed of a set of micro-activities).

Humans construct relationships between function, behaviour and structure through experience and through the development of causal models based on interactions with the artefact. Function is ascribed to behaviour by establishing a teleological connection between the human's goals and measurable effects of the artefact. There is no direct relationship between function and structure (de Kleer and Brown 1984). Behaviour is derived from structure using physical laws or heuristics. This often requires knowledge about external (exogenous) effects and their interaction with the artefact's structure. For example, in a physical manufacturing process, compliance with specified surface tolerances is a behaviour derived from the surfaces achieved (that is an output of the process) and the tolerances given (that represent external benchmarks).

An FBS View of Design Methods

The *Merriam-Webster* dictionary defines a method as "a way, technique, or process of or for doing something". This definition accounts for two aspects that correspond to a method's function and structure, respectively:

- *Method function*: represents the purpose or usefulness of a method "for doing something".
- *Method structure*: represents the internal composition of a method in terms of a "way, technique, or process".

Method function and method structure are addressed in most work on design methodology (even though the terms used for describing them often differ). They can be used as a basis for selecting design methods (Franke and Deimel 2004). This Section presents method function and structure in more detail, and adds method behaviour as a third important aspect of design methods.

Method Function

Important functions of design methods are those concerned with providing support for "doing designing" (a specialised class of "doing something", see *Merriam-Webster*'s definition). A number of process frameworks of designing have been proposed that can be viewed as high-level design methods, described at varying levels of detail and domain-specificity (e.g., Hubka and Eder (1996), Pahl and Beitz (2007), and Gero and Kannengiesser (2004)). Every component (activity) described in these methods can again be viewed as an instance of "doing something", and can thus provide the basis for specifying sub-functions to be fulfilled by more fine-grained design methods. This results in hierarchies of design methods at different levels of abstraction. For example, Hubka and Eder's (1996, p. 135) distinction between "design stages", "design operations", "basic operations", "elementary activities" and "elementary operations" can be used as a basis for constructing such a hierarchy. Method functions provide meaningful labels for indexing individual methods (Chandrasekaran et al. 1998).

Functions that are universal to all design methods include repeatability and reproducibility. Although often not explicitly stated, these functions establish the precondition for identifying and extracting a method as a reusable entity from otherwise transitory streams of design actions.

Method Behaviour

The notion of method behaviour is often neglected in descriptions of methods in the literature. This is because behaviour deals with measurable criteria for evaluating method performance that in most cases can be derived only for specific instances of design methods during or after their use for a given design problem. However, behaviour can be specified as empirical measures of expected or "actual" performance based on experiences with multiple instances of the method. An example is the concept of precision, which is a behaviour required to achieve the functions of repeatability and reproducibility of the method. It can be specified quantitatively in terms of the standard deviation of the results produced by using the method, or qualitatively using labels such as "low" or "high". Precision is derived from the method's structure and its interaction with the method user's experience and understanding of the design problem.

Method Structure

The structure of a design method is best understood as a process. Processes can be looked at from various perspectives, most of which can be grouped into one of four categories (Curtis et al. 1992): the "task", the "workflow", the "organisational" and the "informational" perspective[1]. Table 1 shows how these perspectives map onto different aspects of method (process) structure in the FBS ontology.

Table 1. Mapping four process perspectives onto method structure

Aspects of method structure in the FBS ontology	Process perspectives (Curtis et al. 1992, p. 77)
i (elementary) t (elementary) o (elementary)	*Task Perspective*: "what process elements are being performed, and what flows of informational entities (e.g., data, artefacts, products), are relevant to these process elements"
t (decomposed into flows of activities)	*Workflow Perspective*: "when process elements are performed (e.g., sequencing), as well as aspects of how they are performed through feedback loops, iteration, complex decision-making conditions, entry and exit criteria, and so forth"

[1] Curtis' original terms for the "task" and the "workflow" perspective (namely "functional" and "behavioural", respectively) have not been adopted in this paper to avoid confusion with the notions of function and behaviour in the FBS ontology.

object-centred and process-centred micro-structure of i, t and o	*Organisational Perspective*: "where and by whom (which agents) in the organisation process elements are performed, the physical communication mechanisms used for transfer of entities, and the physical media and locations used for storing entities"
i (decomposed into information structures) t (decomposed into flows of information) o (decomposed into information structures)	*Informational Perspective*: "the informational entities produced or manipulated by a process; these entities include data, artefacts, products (intermediate and end), and objects; this perspective includes both the structure of informational entities and the relationships among them"

The different perspectives shown in Table 1 are fundamental in design methodology, as we will show in the next Section.

Fundamental Approaches

Five fundamental approaches can be derived from the perspectives of method structure. They are referred to as black-box, procedural, artefact-centric, formal, and managerial approaches. However, it is important to note that some instances of methods may map onto more than one approach.

Black-Box Approach

This approach adopts the task perspective. Every task is specified only by its input, transformation and output. In most cases, only the top-level task is specified; any lower-level tasks are not shown and left inside the "black box". Most black-box descriptions of a method do not clearly separate the three components of a task, referring to them by a single label constructed as a verb-noun phrase, for instance "finalise details". This label can sometimes be very similar to the one denoting the function of the method, which is a frequent cause for confusion.

The black-box approach is typically used in two circumstances: (1) when the transformation specified by the method can be performed by an elementary activity, or (2) when little is known about the detailed activities needed to perform the transformation. For example, Hubka and Eder's (1996) "elementary operations" in designing include both common-sense activities such as "see", "read" and "listen", and more complex activities such as "synthesise" and "induct". In both cases, the method serves only as a role description of a potential method user, but provides very limited guidance on how to fulfil this role. In essence, this approach assumes that the method resides within a user who is sufficiently skilled to perform the method. This positions the black-box approach at the lower bound of what can be validly termed a method.

Procedural Approach

This approach adopts the workflow perspective, which is the most common interpretation of a method. Here, the method's structure is described by a

sequence of activities or steps, resembling a recipe or plan. This approach is usually reflected in terms such as "procedure", "technique" and "process". It usually does not specify whether the individual activities are executed by human operators or computational tools. Figure 1 shows an example of the procedural approach.

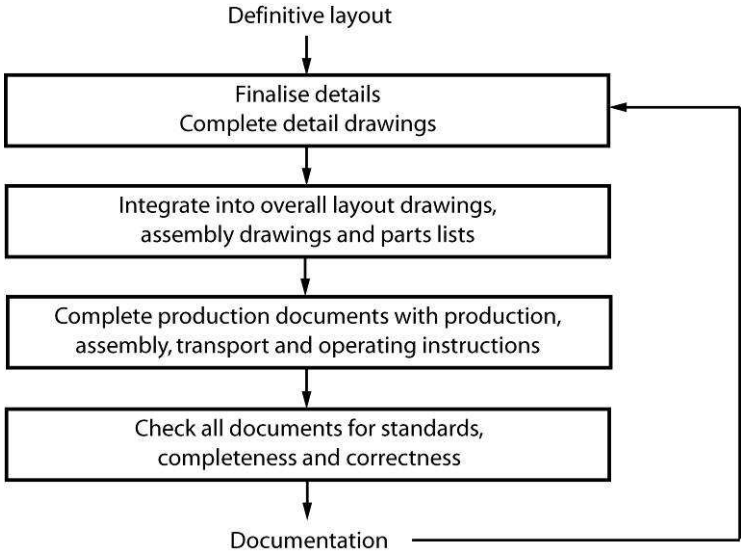

Figure 1. Example of a procedural approach: Detailing a design (after Pahl and Beitz (2007))

Note that every activity within the procedural description can be viewed as an individual (sub-) method. At the level of granularity depicted in Figure 1, they appear as black-box methods. However, it is possible to "explode" their representation to reveal further details that may then be consistent with one of the other approaches.

Artefact-Centric Approach

This approach adopts the informational perspective, emphasising representations of the artefact. The difference between procedural and artefact-centric method approaches is similar to distinctions made by Finger and Dixon (1989) between a "canonical design process" and a "prescriptive model of the design artefact", and by Browning et al. (2006) between activity-based and deliverable-based process models. Artefact-centric representations focus on generic or specific aspects of an artefact and their relationships, often consisting of guidelines, checklists and tables. This approach is often alluded to when using terms such as "notation" and "(object) model". Table 2 shows an example of a method based on generic artefact descriptions used for morphological analysis (Zwicky 1948). More specific artefact-centric methods have been described in design catalogues (Roth 1982), principles or guidelines for embodiment design (Hubka 1982; French 1988; Pahl and Beitz 2007), design patterns (Gamma et al. 1995) and functional taxonomies (Szykman et al. 2001). For example, a design principle by French (1988, p. 195) states that "when guiding one body relative to another, or securing one body to another, use the least number of constraints that will do".

Table 2. Example of an artefact-centric approach: Morphological matrix

Functions	Solution 1	Solution 2	Solution 3	Solution 4
F_1	S_{11}	S_{12}	S_{13}	S_{14}
F_2	S_{21}	S_{22}	S_{23}	S_{24}
F_3	S_{31}	S_{32}	S_{33}	S_{34}
F_4	S_{41}	S_{42}	S_{43}	S_{44}
F_5	S_{51}	S_{52}	S_{53}	S_{54}

A number of methods include both artefact-centric and procedural elements. These elements are not always clearly separated, and are frequently integrated in the natural-language labels of some of the method's activities. For example, Cross (2000) describes morphological analysis as a sequence of activities whose labels subsume artefact-centric representations: (1) "List the features of functions that are essential to the product", (2) "for each feature or function list the means by which it might be achieved", (3) "draw up a chart containing all the possible sub-solutions", and (4) "identify feasible combinations of sub-solutions" (Cross 2000, pp. 124-125).

Formal Approach

This approach adopts the organisational perspective, assuming a computational tool as the agent performing the method. Often, the term "tool" is used for referring to this approach. The sequence of activities and artefact representations dealt with by the tool are omitted, as they are not directly relevant to the method user as long as the tool is available and delivers the results expected. In some sense, this approach has a black-box flavour (unless it is viewed by a method engineer who is interested in the procedural or artefact-centric details). However, it is important to note that the complex, internal details of the formal method are only hidden for user convenience. This is in contrast to the black-box approach where the details are either too trivial or too unknown to be represented explicitly. Common examples of formal approaches to engineering design methods include computer-aided design (CAD), computer-aided engineering (CAE) and computer-aided manufacturing (CAM) tools.

Managerial Approach

This approach adopts the organisational perspective as well, but uses a broader view of the design agent as a system of interactions between human designers, tools and documents. This system is described as a framework of processes that direct, coordinate and control the interactions. The managerial approach maps onto what Hubka (1982) refers to as "working principles" that "give general instructions for appropriate behaviour for the designer" (Hubka 1982, p. 40). The basic assumption is similar to the black-box approach: the potential for performing a particular design activity (i.e., for achieving the method function) resides or emerges within the human designer as some form of "implicit method". The managerial approach aims to unlock this potential by creating a controlled environment that is presumed to facilitate or promote the desired effects on the designer's behaviour. An example of this approach is the brainstorming method (Osborn 1963) that is a coordination process aimed at stimulating the generation of ideas. Another example is

Hubka's (1982, p. 40) general "principle of recording information" that states that "every important item of information should be recorded and classified in an economic fashion".

What Approach for What Design Activity?

We can correlate the fundamental approaches to describing method structure with particular classes of design activities (i.e., functions) to be supported by methods. A comprehensive framework of generic activities in engineering design with mappings to some common methods has been proposed by Sim and Duffy (2003). We can expand this work by including additional methods from standard literature in engineering design, and by identifying their fundamental approaches, Table 3. The design methods are shown as references to the literature, using acronyms that are defined in Table 4. The method functions correspond to what Sim and Duffy (2003) refer to as "design definition activities" and "design evaluation activities". The order in which the five fundamental approaches are presented in this Table (i.e., from left to right) indicates increasing degrees of presumed technological maturity. This is based on the different assumptions of the approaches regarding the involvement of human expertise.

We can see that the black-box approach is used for design activities that can be viewed as elementary (defining, standardising and decision making), involving domain expertise (decision making) or not being at the centre of interest of design methodology (testing/experimenting). If more guidance is needed in performing these activities, a review of more specialised literature may open up some of these "black boxes" to reveal more details. (Such a review is beyond the scope of this paper.)

On the other end of the spectrum is the formal approach. Here, tools are provided with detailed instructions to automatically perform the right tasks at the right time. Table 3 shows that this approach is used for some activities of analysing, modelling and simulating. These activities embody all the domain knowledge and task knowledge required to perform the method.

The managerial approach is used for the activity of associating. This activity is most closely related to creating novel design concepts as required for non-routine designing. The creative ability is generally assumed to reside within the human designer. The managerial approach can be very effective, but its outcomes are often poorly reproducible. While some management support tools are available for this approach, there is no direct technological support for the creative transformation.

The artefact-centric approach is used for a wide range of design activities and can operate on specific as well as general representations of the artefact. This approach often requires human expertise for transforming these representations. Its direct contribution to supporting non-routine design activities is fairly small. For example, checklists used for supporting associating can be regarded as a set of stimuli to a human individual that inspire rather than determine the generation of new design ideas.

The procedural approach is similar to the artefact-centric approach in that its application range is rather large. It requires human expertise for applying the method to appropriate representations of the artefact. As a result, the

procedural approach is sometimes used in conjunction with the artefact-centric one. Procedural methods are primarily used for coarse-grained design activities such as synthesising and decomposing.

Table 3. Engineering design methods mapped onto generic method functions (based on Sim and Duffy (2003)) and fundamental approaches to method structure

Method Function	Black-Box	Managerial	Artefact-Centric	Procedural	Formal
Synthesising			schematic synthesis (UlrSee89) functional synthesis (ChaBli94)		
			design for X (DFX) (Bra96)		
Abstracting				PahBei07, p.165	
Generating			working principles (PahBei07, pp.181-186) mapping (Suh90)	morphological analysis (Cro00, pp.124-125)	
Decomposing			decomposition by function (PahBei07, pp.169-181; Suh90)	function analysis (Cro00, p.81) establishing the function structure (HubEde96, p.136) decomposition by product modularity (KusWan93)	
Associating		brainstorming (Osb63) synectics (Gor61)	checklist for idea generation (ThoLor99)		
Composing			combining working principles (PahBei07, pp.184-186)		
Structuring/ integrating				integration analysis (PimEpp94)	
Detailing			principles & guidelines (PahBei07; Fre88)	PahBei07, p.437 HubEde96, p.136	
Defining	SimDuf03				
Standardising	SimDuf03				
Decision making	SimDuf03				
Evaluating			checklist (PahBei07, p.193 & 416) screening matrix (UlrEpp95)	weighted objectives (Cro00, pp.140-147; PahBei07, pp.109-123)	
Selecting			selection chart (PahBei96, p.108)	PahBei96, pp.106-109	
Analysing				SimDuf03	SimDuf03
Modelling					SimDuf03
Simulating					SimDuf03
Testing/ experimenting	SimDuf03				

Table 4. Definitions of acronyms used in Table 3

Acronym	Reference
Bra96	Bralla 1996
ChaBli94	Chakrabarti and Bligh 1994
Cro00	Cross 2000
Fre88	French 1988
Gor61	Gordon 1961
Hub82	Hubka 1982
HubEde96	Hubka and Eder 1996
KusWan93	Kusiak and Wang 1993
Osb63	Osborn 1963
PahBei07	Pahl and Beitz 2007
PimEpp94	Pimmler and Eppinger 1994
SimDuf03	Sim and Duffy 2003
ThoLor99	Thompson and Lordan 1999
UlrEpp95	Ulrich and Eppinger 1995
UlrSee89	Ulrich and Seering 1989

Dimensions of Prescriptiveness

Design methods can be viewed as artefacts that traverse a life cycle of development, implementation, execution, assessment and disposal (de Araujo 1996). The role of designers is generally understood as the "users" of these artefacts. However, this role is much more complex than being the user of artefacts such as cars, buildings and mobile phones. These artefacts readily afford specific user behaviour without involving significant reasoning effort. In contrast, using design methods frequently requires elaborating, combining and modifying these methods to fit with the individual design problem. This entails using considerable amounts of experience and can even be viewed as an act of (re-) designing rather than merely using a method (van Aken 2005). The well-known phrase "the script is not the play" can be used as a metaphor for the difference between a design method and the "actual" course of design actions.

On the other hand, no one would argue that a "script" or method is unnecessary. Design methods can provide useful guidance for meeting goals and constraints that an individual designer may not be fully aware of. A method given to a designer can be viewed as a requirements artefact that constrains the designer's actions in a purposeful way. Prescriptiveness is a property that measures the extent to which these requirements are binding and set limits for the designer's "realisation freedom". This Section describes prescriptiveness as an aggregate construct that can be characterised along four dimensions: certainty, granularity, flexibility and authority. Every dimension is described based on the FBS ontology of design methods, Figure 2.

Certainty: Prescribing Method Function, Behaviour or Structure

We have shown that design methods can be described at three levels: function, behaviour and structure. Methods that are not described at all three levels can be termed ontologically incomplete (Wand and Weber 1993). Ontological incompleteness of a method specification frequently occurs in the early stages of the method life cycle, usually at the levels of structure and, sometimes, behaviour. In other words, while it is usually specified what the method is for (function) and what performance criteria are relevant (behaviour), not all aspects of structure may be known prior to the method's realisation. The extent to which method structure and behaviour (besides function) are specified at the outset of method use can be called *certainty*.

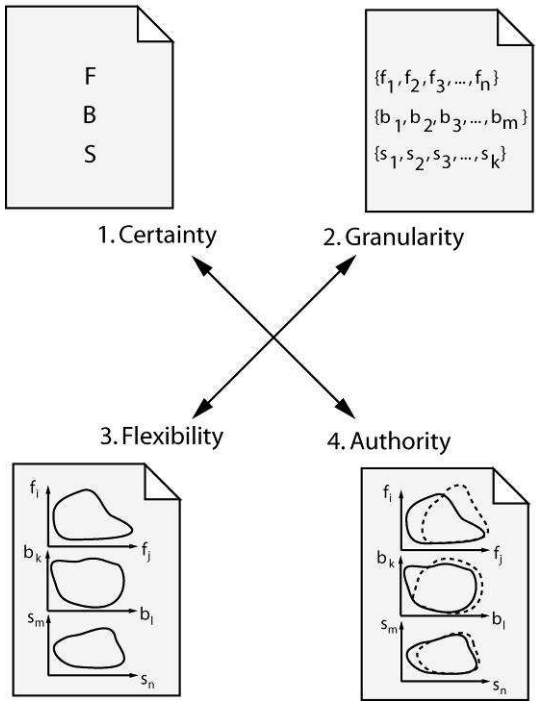

Figure 2. Four dimensions of prescriptiveness

Granularity: Prescribing Method Variables

Design methods can be specified at varying levels of detail or *granularity*. This notion can be viewed as the location where the micro-structure of a method is distinguished from its macro-structure. Granularity is also the determinant of whether an individual activity is viewed as a "black box" (i.e., elementary) or as a function to be achieved by more fine-grained (yet unknown) activities. The main factor for selecting an appropriate granularity for a method is the degree of difficulty associated with the design activities to be supported, which depends on the nature of the design task and the available knowledge representation. Typically, the level of granularity increases as the method progresses through its life cycle.

Flexibility: Prescribing Method Variants

The description of a design method can specify which method variants (if any) are permitted. The set of all variants can be termed the state space of possible (or permissible) design methods. In fact, the view of method use as an instance of (re-) designing (van Aken 2005) allows viewing this state space as a design state space. A design state space has three subspaces: a function state space, a behaviour state space and a structure state space. The ranges of values specified for the individual dimensions of a method design state space determine the *flexibility* of the method. The broader these ranges are, the more variants are allowed and thus the more flexible is the method. Flexibility in method descriptions is often provided for method behaviour. For example, time constraints can be specified that allow for method variants with speeds faster than a required minimum value. An example of flexible method structure is the specification of a maximum number of iterations allowed within the transformation.

Authority: Prescribing the Potential for Method Reformulation

Requirements in design are sometimes viewed as "hard" (mandatory) or "soft" (optional or desirable). The same distinction can be applied to method artefacts. The notion underpinning this concept can be termed *authority*. It reflects the organisational and socio-cultural context of method use, which may be pre-defined or emerge as a result of negotiation between the stakeholders. Authority is required whenever the state space of a method needs to be modified beyond the specified bounds of flexibility, by changing the set of method variables or their ranges of values. Modifications of this kind can be called method reformulation.

Conclusion

The ontological basis proposed in this paper enhances understanding of design methods by addressing the two issues outlined in the introductory part of the paper.

Terminological issues have been shown to be based on different process perspectives that can be interpreted as different approaches to method structure. Five fundamental approaches have been identified that characterise design methods independently of the design domain and the specific terms and concepts used. This paper has demonstrated that a number of methods in the domain of engineering design can be classified according to this schema. The correlations established with various classes of design activities show that the different approaches can be interpreted as indicators for the technological maturity of a method. Further research may refine these indicators by integrating the interaction of methods with exogenous effects, including different types of method users. Our ontological basis allows applying such a study to other design domains. Interesting target applications include some of the emerging design disciplines, such as business process design and interaction design.

Prescriptiveness has been specified as a four-dimensional construct rather than a simple classifier of the binary "prescriptive vs. descriptive" distinction. This facilitates the management of design projects by providing the basis for

exact descriptions of the way in which design methods are to constrain a designer's actions. Our notion of prescriptiveness is founded on a view of design methods as external requirements on design actions, to be communicated to a designer. This view relates our work to previous research by Stacey and Eckert (2003) on potential forms of ambiguity in design communication. Specifically, our dimension of flexibility maps onto their notions of precision and sensitivity, and our dimension of authority maps onto their notion of commitment.

Viewing methods as artefacts that are represented using the FBS ontology opens up at least two research avenues. First, representation languages of design methods can be developed that provide explicit, formal constructs for specifying prescriptiveness along all dimensions. Currently, most design methods are only informally represented and do not fully support the four dimensions. For example, method flexibility is not well supported, due to the lack of declarative languages for specifying explicit constraints on method structure.

The second research avenue is a further investigation of the idea of method use as a (re-) design process. The situated FBS framework that represents the activities involved in situated designing (Gero and Kannengiesser 2004) can be used for describing the interaction between an externally specified method and the designer's internal interpretations and expectations of that method. This may lead to the identification of new research issues related to the use of design methods, derived from known phenomena in traditional design domains such as architecture and engineering. Possible issues include method fixation and method emergence.

Most importantly, the ideas presented in this paper can serve as a framework for research within and across disciplinary boundaries, no matter where these boundaries are located, which areas of design they delineate, and how recently they have been drawn. This is based on the uniformity with which all design methods are represented, independently of the specific discipline, school of thought or level of detail. Using a design ontology such as the FBS ontology makes a number of concepts that are already known in the world of designing accessible for the world of design methodology. In particular, the notions of function and behaviour promote a unified view of rigour of inquiry in design methodology, as they capture important concepts such as usefulness (function) and measures for evaluating quality and performance (behaviour).

Acknowledgements

NICTA is a national research institute with a charter to build Australia's pre-eminent Centre of Excellence for information and communications technology (ICT). NICTA is building capabilities in ICT research, research training and commercialisation in the ICT sector for the generation of national benefit. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

van Aken, J.E. (2005) Valid knowledge for the professional design of large and complex design processes, *Design Studies* **26**(4): 379-404.

de Araujo, C.S. (1996) Sharpening understanding of design methods, *in Proceedings of the 1st Annual International Conference on Industrial Engineering Applications and Practice*, Houston, TX.

Bralla, J.G. (1996) *Design for Excellence*, McGraw Hill, New York.

Browning, T.R., Fricke, E. and Negele, H. (2006) Key concepts in modeling product development processes, *Systems Engineering* **9**(2): 104-128.

Chakrabarti, A. and Bligh, T.P. (1994) An approach to functional synthesis of solutions in mechanical conceptual design. Part 1: Introduction and knowledge representation, *Research in Engineering Design* **6**(3): 127-141.

Chandrasekaran, B., Josephson, J.R. and Benjamins, V.R. (1998) Ontology of tasks and methods, *in Proceedings of the 11th Knowledge Acquisition Modeling and Management Workshop (KAW'98)*, Banff, Canada.

Cross, N. (2000) *Engineering Design Methods: Strategies for Product Design*, John Wiley & Sons, Chichester.

Curtis, B., Kellner, M.I. and Over, J. (1992) Process modeling, *Communications of the ACM* **35**(9): 75-90.

Eekels, J. and Roozenburg, N.F.M. (1991) A methodological comparison of the structures of scientific research and engineering design: their similarities and differences, *Design Studies* **12**(4): 197-203.

Finger, S. and Dixon, J.R. (1989) A review of research in mechanical engineering design. Part I: Descriptive, prescriptive, and computer-based models of design processes, *Research in Engineering Design* **1**(1): 51-67.

Franke, H.-J. and Deimel, M. (2004) Selecting and combining methods for complex problem solving within the design process, *in* D. Marjanovik (ed.) *Design 2004*, Dubrovnik, Croatia, unnumbered.

French, M.J. (1988) *Invention and Evolution: Design in Nature and Engineering*, Cambridge University Press, Cambridge.

Frey, D.D. and Dym, C.L. (2006) Validation of design methods: Lessons from medicine, *Research in Engineering Design* **17**(1): 45-57.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software,* Addison-Wesley, Boston.

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

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

Gero, J.S. and Kannengiesser, U. (2007) A function-behavior-structure ontology of processes, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **21**(4): 379-391.

Gordon, W.J.J. (1961) *Synectics*, Macmillan, London.

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

Hubka, V. and Eder, W.E. (1996) *Design Science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge*, Springer-Verlag, Berlin.

de Kleer, J. and Brown, J.S. (1984) A qualitative physics based on confluences, *Artificial Intelligence* **24**: 7-83.

Kusiak, A. and Wang, J. (1993) Efficient organizing of design activities, *International Journal of Production Research* **31**(4): 753-769.

Osborn, A.F. (1963) *Applied Imagination*, Scribners, New York.

Pahl, G. and Beitz, W. (2007) *Engineering Design: A Systematic Approach*, Springer-Verlag, Berlin.

Pedersen, K., Emblemsvag, J., Bailey, R., Allen, J.K. and Mistree, F. (2000) Validating design methods & research: The validation square, *in Proceedings of DETC'00 2000 ASME Design Engineering Technical Conferences*, Baltimore, MD.

Pimmler, T.U. and Eppinger, S.D. (1994) Integration analysis of product decompositions, *in ASME Design Theory and Methodology (DTM'94)*, pp. 343-351.

Roth, K. (2000) *Konstruieren mit Konstruktionskatalogen: Band 2: Kataloge*, Springer-Verlag, Berlin.

Sim, S.K. and Duffy, A.H.B. (2003) Towards an ontology of generic engineering design activities, *Research in Engineering Design* **14**(4): 200-223.

Stacey, M. and Eckert, C. (2003) Against ambiguity, *Computer Supported Cooperative Work* **12**(2): 153-183.

Szykman, S., Fenves, S.J., Keirouz, W. and Shooter, S.B. (2001) A foundation for interoperability in next-generation product development systems, *Computer-Aided Design* **33**(7): 545-559.

Thompson, G. and Lordan, M. (1999) A review of creativity principles applied to engineering design, *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* **213**(1): 17-31.

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

Ulrich, K.T. and Seering, W.P. (1989) Synthesis of schematic descriptions in mechanical design, *Research in Engineering Design* **1**(1): 3-18.

Vermaas, P.E. and Dorst, K. (2007) On the conceptual framework of John Gero's FBS-model and the prescriptive aims of design methodology, *Design Studies* **28**(2): 133-157.

Wand, Y. and Weber, R. (1993) On the ontological expressiveness of information systems analysis and design grammars, *Journal of Information Systems* **3**(4): 217-237.

Zwicky, F. (1948) *Morphological Analysis and Construction*, Wiley Interscience, New York.

Udo Kannengiesser

Udo Kannengiesser is a Researcher in the Software Engineering research team at NICTA, Australian's Centre of Excellence for information and communication technology. His current research focuses on modelling processes in design, manufacturing, business and government. Before joining NICTA, Udo Kannengiesser was a research assistant at the Key Centre of Design Computing and Cognition (University of Sydney), where he also obtained his PhD. His doctoral studies were concerned with a number of topics in design research, including situatedness in design, the FBS ontology, and agent-based models of interoperability. Udo Kannengiesser also holds a degree in mechanical engineering from the University of Karlsruhe (Germany) with a specialization in information systems for design and production.