COMPARISONS OF DESIGN THEORIES

W. Ernst Eder and Ch. Weber

Keywords: Design theories

1. Introduction

In recent years, many theories have been proposed to describe ‘design’, ‘design processes’ and other similar concepts. Various published design methodologies may be regarded in one sense as theories of design, e.g. Pahl [2006], Koller [1985], Roth [1995], Dietrych [Hubka 1982a], VDI [1977]. Many of these are partial theories, some confuse ‘design’ as a noun with ‘design’ as a verb, and all can be seen as part of the Engineering Design Science [Hubka 1992b, 1996]. The Theory of Technical Systems, TTS [Hubka 1984 and 1988], presenting a science of object knowledge, and the Theory of Design Processes [Hubka 1976], presenting a science of design process knowledge, provide the theoretical basis for Engineering Design Science, figure 1.

![Figure 1. Model (Map) of Engineering Design Science](image)

2. Procedural Model
Andreasen [1980], after close cooperation with Hubka, proposed a ‘domain theory’, based directly on the structures of technical systems, TS [Hubka 1974, 1976 and 1984]. Each TS-structure has a ‘domain’ on orthogonal axes of ‘abstract to concrete’ and ‘incomplete to complete’, designing aims towards concrete and complete description of a TS(s). A further development of a ‘function-means tree’ is a reflection of the scheme of ‘goals-means’ [Hubka 1984, fig. 5.13, p. 78], see figure 2, and depicts the sequencing of steps for any recognized evoked functions. An extension of the ‘chromosome model’ by Mortensen [1999] shows the relationship among the TS-structures, as shown in a text passage in [Hubka 1984, fig. 5.4, p. 60-61]. These partial design theories are graphic clarifications, and are a sub-set of TTS.

Figure 2. Scheme ‘Goals – Means’ or ‘Effects – TS

Albers [2003 and 2004] proposed the ‘contact and channel model’, C&CM. Contacts are defined as ‘working surface pairs’, identical to ‘organs’ [Hubka 1984 and 1988], channels are defined as ‘support structures’, identical to ‘constructional parts’ [Hubka 1984 and 1988]. The C&CM model by Albers seems to apply mainly to mechanical systems, it is thus also a sub-set of TTS, but this is not acknowledged by Albers.

Pahl [2006] and VDI 2225:1975, VDI 2222:1977 and VDI 2221:1985 show a procedural model of design engineering based on pragmatic considerations. All the steps in this model are included in the procedural model of Engineering Design Science [Hubka 1992b, 1996]. In the VDI model, a ‘total function’ for a technical system is defined, which includes the transformation process, TrfP, as shown in figure 3. This ‘total function’ is then ‘decomposed’ to the TS-internal functions. We prefer a complete separation of TrfP and TS, which
consequently allows and encourages consideration of all operational states and ‘duty cycles’ of the TrfP(s) and of the TS(s). This separation also encourages a consistent view of all engineering design problems at any level of complexity, and shows that using a TS-internal function from a higher-level view can be used as transformation process for a lower-level.

Figure 3. General Model of Transformation System

The German Society for Engineers has recently published a guideline for mechatronic systems, VDI 2206:2004, which includes a ‘V-model’ of design development, figure 4. By implication, the procedural models of VDI 2225:1975, VDI 2222:1977 and VDI 2221:1985 are included in the ‘domain-specific design’. Blanchard [2004] shows a similar model with respect to software systems. We claim a similarity to the Procedural Model of Design Engineering [Hubka 1992b, 1992a, 1996]:

- the ‘domain-specific design’ is represented by separate functions in the function structure, which may specify functions that can be realized by mechanical, electrical, chemical, software, or any other system,
- ‘integration’ can and should take place in any of the relevant structures (TrfP, TgStr, FuStr, OrgStr, CStr), but is especially necessary in the constructional structure because cooperation among the specialists is especially necessary here, and
- the cycle of ‘substantiate, verify, improve’ at the end of each design stage in the Procedural Model [Hubka 1992a, 1992b, 1996] leads to a feedback to any previous stage, not just to the horizontally referenced level, although this level may be the most likely target.

Figure 4. ‘V’-Model of Design Development
3. Pseudo-Theories

TRIZ and its equivalents stems from an extensive investigation by Altshuller [1973 and 1987], who searched several thousand patents to discover parameters and principles for technical systems, presented as a ‘Theory of the Solution of Inventive Problems’. He proposed a method to develop clever solutions to problems that show a contradiction, where improving one parameter would adversely influence another parameter. 39 ‘general parameters’ (equivalent to some of the TS-properties) were defined, and 40 ‘principles’ for finding design solutions were found, but neither include the electronic, digital-electronic and mechatronics principles. At least, they have more recently been put into English-language terms, not just straight translations from the Russian. Coherent theories do not exist for object-related information, nor for design processes, and the two lists of ‘parameters’ and ‘principles’ are obviously neither complete nor logical in their arrangement.

Axiomatic Design was proposed by Suh [1989], but note that no advice is given by Suh about performing the design process to establish candidate solutions, he declares this as simply ‘creative’. Suh defines design as a mapping of FRs – functional requirements, to proposed solutions, DPs – design parameters in the physical space. He acknowledges further mappings from the customer space to the functional space, and from the physical space to the process domain of manufacturing. Each of the FRs and DPs is assumed to behave in a linear fashion. If the numbers of FRs and DPs can be made equal, a square matrix of FRs vs. DPs can be formulated, which can be inverted – implying that synthesis is a direct inversion of analysis, but this is necessarily a special case. Analysis is in essence a one-to-one transformation, and is in some ways a reversal of synthesis. Synthesis goes far beyond a reversal of analysis, it is almost always a transformation that deals with alternative means and arrangements, a one-to-many (or few-to-many) transformation. Synthesizing is the more difficult kind of action. The axioms and procedures are intended for evaluation of the ‘proposed designs’ (noun), making decisions about the ‘best’ of the candidates according to mathematically solvable criteria can then be performed by linear algebra, i.e. matrix methods.

- **Axiom 1**: The Independence Axiom – Maintain the independence of FRs – all functional requirements are preferably assumed orthogonal to each other, interactions are to be avoided.
- **Axiom 2**: The Information Axiom – Minimize the information content.
- Eight ‘Corollaries’ and 16 ‘Theorems’ complete the listing.

This normally leads to formulating complex FRs, and probably simplistic choices [Starr 1963, Morrison 1968]. The simplistic mapping of FRs to DPs by Suh, with no search for alternatives, may be compared with the multiple mappings recommended in [Hubka 1992a, 1992b, 1996], in which alternative solutions can be developed: design specification – transformation process TrP – technologies TgStr – function structure FuStr – organ structure OrgStr – constructional structure CStr in preliminary layout – definitive layout – detail, steps in the Procedural Model.

4. Set Theoretic Models

The General Design Theory, GDT, was proposed by Yoshikawa [1981a, 1981b, 1981c and 1983]. It is based on a mathematical set-theoretic and deterministic world view in which the ‘ideal knowledge’ includes everything that is now known, and everything that will be known in future. GDT only considers a technical system once it exists, with a one-to-one mapping of entities onto their representations (concepts). There is no envisaged possibility of searching for alternative solutions at any level, all possible solutions are already available for selection. In essence, only the final constructional structure is considered, and only those properties that have a measure and value can be included – appearance seems to be denied. The point of overlap between GDT and TTS is the definition of classes of TS-properties. Aims of GDT include absolute optimization, and construction of a computer system and its formulation for computer-aided design. Under these conditions, synthesis is a direct matrix inversion of analysis, and the full ‘design intent’ should be available for capture by computer processing. In this sense, the programs of Product Data Management are similar.
Tomiyama [1995] has extended GDT to include only his own cognitive research results, see also [Yoshioka 1999], to produce a ‘theory of synthesis’. The resulting ‘cognitive design process model’ shows some similarity to the problem solving process in figure 5. Its implementation in a CAD program seems to have been achieved. The human capacity for novel and associative thinking seems still to be largely ignored.

Lossack [2002a], under supervision from Grabowski, proposed a Universal Design Theory (UDT) based on a methodological framework consisting of ‘theory’, ‘applications’ and ‘validation’ to characterize a ‘design working space’ in preparation for computer processing. ‘Theory’ is divided into ‘solution patterns’ of design knowledge, and a ‘formal framework’ containing design guidelines, design principles, and axioms. ‘Applications’ are claimed from mechanical engineering, chemistry, materials science, computer science, biology, pharmacology, and architecture. ‘Validation’ should be by empirical research, utilization and transfer. The design process is described using the methods and ‘layered model’ of VDI 2221:1992 and VDI 2222:1977 coupled with a generic problem solving cycle developed by Rutz [1985]. The resulting connections among the ‘requirements’, ‘function’, ‘physical principle’ and ‘embodiment’ layers looks strangely like the chromosome model of Andreasen [1980], but with a better formalization of the relationships. Lossack [2002b] expanded UDT by attempting to define a Domain Independent Design Theory (DIDT).

Grabowski [2004] reported an attempt to use UDT to create a computer program for ‘requirements development’. The non-deterministic requirements development process was divided into elemental steps, and described by its states, and the appropriate state transitions. This process should result in a progressively more detailed requirements network. Developing the ‘requirements’ allows selection of constructional parts for the product. A software prototype was produced. It seems that the authors in part confused ‘requirements’ (normally pre-specified) with TS-functions.

5. AI Applications

Hatchuel [2003 and 2006] proposed a new unified C-K theory of design that tries to avoid the restrictions of GDT and UDT. Their survey of existing theories does not include the works of Hubka and associates. There seems to be no differentiation between information (including knowledge and data) that is internalized in mental structures of humans, and information that is available in recorded form. Design should be defined independent of any domain or
professional tradition – which seems to deny any differences between design engineering and the more artistic design disciplines, see figure 6. ‘K’ is defined as a knowledge space, containing propositions that have a logical status for the designer – logical status defines the degree of confidence that a designer assigns to a proposition. ‘C’ is defined as a concept space, in which the propositions have no logical status – does this mean that they are illogical, or only that they are in human minds? Apparently, the only operations that can be performed are K→C, C→K, C→C and K→K. By definition, ‘design’ is a process of generating other concepts or transforming them into knowledge. How this is to be done is not defined in any way. Hatchuel claims that ‘the metaphors of “exploration” and “search” are confusing for design’, yet we all explore and search for possible solutions from existing precedents [Booker 1962], from tacit/internalized knowing, from the literature, and many other locations – does this mean that the C-K ‘concept space’ cannot exist? And what are the ‘properties’ that can be added or subtracted from the initial ones? Without disputing the claimed rigor of the C-K theory, it seems that this formulation has some similarity with the interaction of cognitive processes (in a human mind) and the external representations produced by a human.

Hatchuel claims that he has a formal proof of correctness, based on set-theory – which is also the basis of GDT by Yoshikawa. He propagates this view only at Computer Science conferences (“because design people would not understand it”), whereas at design conferences he tries to propagate aspects of application.

Gero [2003] proposed a model of ‘situatedness’, especially for design computing and artificial intelligence in relation to architecture, see figure 7. Gero also showed a set of relationships concerning function, behavior and structure. Some of these relationships, see also figure 2, are recognized by us as causal, e.g. structure determines actual behavior. Other relationships are not causal, and must be established by a process of finality.

Smithers [1999], using concepts of AI, proposed to define a structure of knowledge at various levels to enable design, as a Knowledge Level Theory of Design (KLDE), independent of implementation. He defines ‘knowledge’ (in an e-mail to Prof. Christian Weber) as a ‘capacity to act rationally with respect to some class of objects’ – without differentiating whether this knowledge exists in tangible records or in the mind of a human. He defines ‘information’ as the communication of data between (knowledgable) agents – ignoring the fact that much information is available in verbal forms, without numerical values [Constant 1980, Vincenti 1990]. ‘Data’ with values is only obtained by measurement, mathematical derivation, or computation – but apparently not by estimation or assessment by humans. Knowledge has three ‘roles’ and four ‘type relations’, from which he defines 18 types of knowledge used, and 13 types created in designing – there seems to be no way to establish that these classifications are complete. Smithers, in his introduction, states: ‘So far, all of this engineering activity has been carried out in the absence of any usable theory or theories of design process’ – how many of us have been wasting our time? For instance, he quotes

![Figure 6. Scope of Sorts of Designing](image-url)
Hubka 1992b, but not Hubka 1996! It seems that everything must be transformed into a computational realization.

A Situatedness in Designing

B Function – Behavior – Structure (F-B-S)

Figure 7. Situatedness in Designing, and Relationship ‘Function – Behavior – Structure’

Braha [2006] describes a rule-based approach to automating a design task. The paper sets out several ‘facts’ that seem to be arbitrary descriptions of usage for a car, a set of 44 ‘structural attributes’ that represent an incomplete and unsystematic collection of items, and a set of 30 ‘functional attributes’ that are equally incomplete and unsystematic. 38 ‘if-then’ rules are laid out to relate the functional and structural attributes. The reported algorithm can then provide a ‘consistent solution’ to the problem, using a Boolean satisfiability encoding. No specific car is recognizable in the reported ‘solution’, in fact the car now needs to be designed for external appearance and for internal functioning to this set of attributes before any parts of it can be made. The reported algorithm is probably useful for pure configuration products, for which each constructional part (sub-system) has been fully designed, manufactured, and tested, ready for final configuration and assembly.

Following from Gero’s [2004] proposal of situatedness, see figure 7, Kazakçi [2005] finds a need to add spaces of the internal and external world to the Hatchuel C-K Theory. The ‘interpreted world’ of Gero is replaced by the C-K spaces.

A more plausible scheme stems from Kuate [2006], see figure 8, derived from a protocol study, which confirms the ‘windows’ view of Nevala [2005b]. When a designer dives into detail, he/she also recalls relevant general and professional information, e.g. mental models of the surrounding constructional structure. Nevertheless, the designer comprehends the total problem through a restricted ‘window’ [Nevala 2005b], as a design zone, including form-giving zone [Hubka 1992a, 1992b, 1996]. The boundaries of that window are determined by the immediate design task, the personal knowing and the organizational position of the individual, and change from incident to incident.
A more comprehensive scheme was proposed by Eeke ls [1994, Roozenburg 1995], see figure 9. This is part of their ‘Logic of Design’, derived from a combination of design engineering and industrial design. They seem to equate ‘function’ with ‘transformation process’ and ‘functioning’. The partial representation of the ‘cosmonomy’ consists of a set of hypothetical statements such as ‘if A, then B’, under the assumption that reality is likely to behave that way, as a ‘causal model’. A discussion of the ‘logic’ aspects shows the formalization of deduction, induction, reduction/abduction, or innovation [Eekels 2000], all of which are needed for science and for design. The ‘logic of design’ represents a more abstract level of science, probably between the ‘general design science’ and the Engineering Design Science in figure 10.
6. Constructional Structure

Property-Driven Development/Design (PDD) [Weber 2004 and 2005b] distinguishes 'characteristics' – almost coincident with internal properties – and 'properties' – almost coincident with external properties of TS. Physical and/or digital/virtual analysis consists of determining and/or predicting a product's external properties and behavior from the existing internal properties. Synthesis and product development consists of establishing and assigning the product's internal properties from the required external properties. Modeling products and product development processes may be performed by a 'Characteristics-Properties Modeling' (CPM) procedure. The internal properties show a complex relationship to the external properties, compare figure 11. In analysis, these relationships are known and can be determined. In synthesis, 'inverting the relationships' can result in conflicts which must be resolved. External conditions are seen as properties of neighboring systems – 'Design for X' is the process of considering these external conditions when designing a product. 'Design of X' is a process of simultaneous engineering of the external conditions, e.g. the manufacturing system.
A first development cycle consists of four basic steps: (a) in a synthesis step, some internal properties are established from the 'as should be' requirements for external properties, (b) in an analysis step, the resulting 'as is' properties are determined, (c) in an evaluation step, the 'as is' properties are compared to the 'as should be' properties, (d) conclusions are drawn from the comparison, and drive and control the continuing process. The similarity to the problem solving process in figure 5 is obvious, the formalization by Weber is probably an improvement for computer processing. Additional development cycles are needed [Weber 2005b] to resolve conflicts as they arise, and to iteratively and recursively progress towards a final designed solution to the problem. Progressively better simulations are possible using the four-pole modeling methods. This theoretical framework seems to be more applicable to the TS-constructional structure, and less to the more abstract TS-structures. The mathematical formalization [Weber 2005a] may lead to a more rational computer-aided design process, even in the stages of conceptualizing.

The Autogenic Design Theory, ADT [Vajna 2005 and 2006], claims that design is evolutionary. Designers develop over time from 'm-designers' with good education in methodical and systematic design, to 'p-designers' whose procedures are driven by intuition and experience. 'P-designers' achieve results in shorter time, but their processes and results are not transparent and traceable. 'M-designers' learn the methods so well that eventually they do not need to refer to the instructions, they then act intuitively, compare [Müller 1990]. Designed solutions tend to become more complex in time, and with progressive concretization, a procedure called autogenesis within evolutionary theory [Csayi 1988]. ADT uses this analogy for product development, and applies genetic algorithms to drive the evolutionary process. This seems to be mainly applicable to optimization problems of parametrization in the TS-constructional structure, by randomly searching for mutations that produce better performance.

9. Closure
Most 'design theories' seem to be restricted to existing products and 'their design' – the appearance and other properties, to design methods with little theoretical underpinning or to cognitive matters of humans. Engineering design science has attempted to combine these trends.

Figure 11. Relationships Among Classes of TS-Properties

Each external property influences several other external properties, while each internal property influences several internal properties. Relationships between external and internal properties are complex and difficult to resolve.
References

– VDI (1975) VDI Richtlinie 2225, Blatt 1 and 2: Technisch-wissenschaftliches Konstruieren (Technical-Scientific Designing), Düsseldorf: VDI-Verlag
– VDI (1985) VDI Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte (Methodology for Developing and Designing Technical Systems and Products), Düsseldorf: VDI


Altschuller, G.S. (1973) Erfindungen - (K)ein Problem in Russland (Inventing - (Not) A Problem in Russia), Berlin: Verlag Tribüne, translated from a much earlier Russian-language publication


Hubka, V. (ed.) (1982a) WDK 9: Dietrych zum Konstruieren (Dietrych about Designing), Zürich: Heurista


W. Ernst Eder
Professor Emeritus, P.Eng., Dr. h.c. (University of West Bohemia in Pilsen, Czech Republic)
Royal Military College of Canada / Collège militaire royal du Canada (retired)
107 Rideau Street, Kingston, Ontario, Canada   K7K 7B2
E-mail  eder-e@rmc.ca       Phone  x-1-(613) 547-5872