INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016
Dubrovnik - Croatia, May 16 - 19, 2016.

SOLUTION PATTERNS – THEIR ROLE IN INNOVATION, PRACTICE AND EDUCATION

C. Weber and S. Husung

Keywords: design theory, design practice, design education, knowledge re-use, innovation

1. Introduction

The re-use of components, concepts and related knowledge is an important factor in design practice, but also in design education. The term "solution patterns" is frequently used both in academia and practice. However, in Design Theory and Methodology there is no uniform concept of what a solution pattern is; subsequently, the use of the term is diverse. Based on the CPM/PDD approach (Characteristics-Properties Modelling – product-describing aspect; Property-Driven Development/Design – process-describing aspect), this contribution defines the term "solution pattern" as the carrier of the product-related knowledge that can be re-used.
From there, the role of solution patterns in innovation is studied as the core issue of this article. The result is that just two innovation types exist in product development/design: Replacing one or more solution patterns in an existing design (type A) or creating one or more new solution patterns and developing them into a state fit for use in a subsequent design process (type B). Because of their importance, the role of solution patterns (as defined before) in design practice and design education is also briefly studied. Conclusions are drawn for further research topics, and for modifications in design education.
Besides the product-related patterns addressed here, certain activity patterns in the design process could be considered, but are outside the focus of this contribution.

2. Framework: modelling products based on characteristics and properties

The CPM/PDD approach (Characteristics-Properties Modelling – product-describing aspect; Property-Driven Development/Design – process-describing aspect) provides a good base for defining and studying patterns used in design. Therefore, before going into details, a brief introduction to the CPM/PDD approach may be permitted [Weber 2007, 2014].
For the subsequent considerations on solution patterns the product-describing aspect (i.e. the "CPM" part) is the main basis. As already mentioned in the introduction, activity patterns in the design process (i.e. addressing the "PDD" part) exist, but will not be addressed in this contribution.
The CPM/PDD approach is based on the distinction between the characteristics (in German: *Merkmale*) and the properties (*Eigenschaften*) of a product:

- ***Characteristics (C_i)*** are made up of the parts structure, shapes, dimensions, materials and surfaces of a product (*Struktur und Gestalt*, *Beschaffenheit*). They can be directly influenced or determined by the designer.
- ***Properties (P_j)*** describe the product's behaviour (*Verhalten*), e.g. function, weight, safety and reliability, aesthetic properties, but also things like manufacturability, assemblability, testability, environmental friendliness, and cost. They can ***not*** be directly influenced by the designer.

The characteristics are very similar to the "internal properties", as defined by [Hubka and Eder 1996], and to what is called "design parameters" in [Suh 1990, 2001]. The properties are related to "external properties" [Hubka and Eder 1996] and to "functional requirements" [Suh 1990, 2001], respectively. The concept of properties, as used here, is also related to the "affordance" approach by [Maier and Fadel 2001, 2009]. For reasons not to be discussed here, the authors prefer to use the terms "characteristics" and "properties" which go back to [Andreasen 1980].

To handle characteristics and properties – thousands of them in complex products – and to keep track of them in the process they have to be structured. **Figure 1** shows the basic concept:

- On the left, a proposition for the (hierarchical) structuring of characteristics is given, following the parts' structure of a product. It complies with standard practices, and links to common data structures of CAx-systems.
- On the right, a first proposition for the structuring properties is presented, based on life-cycle criteria, and reflecting frequently discussed issues in design.

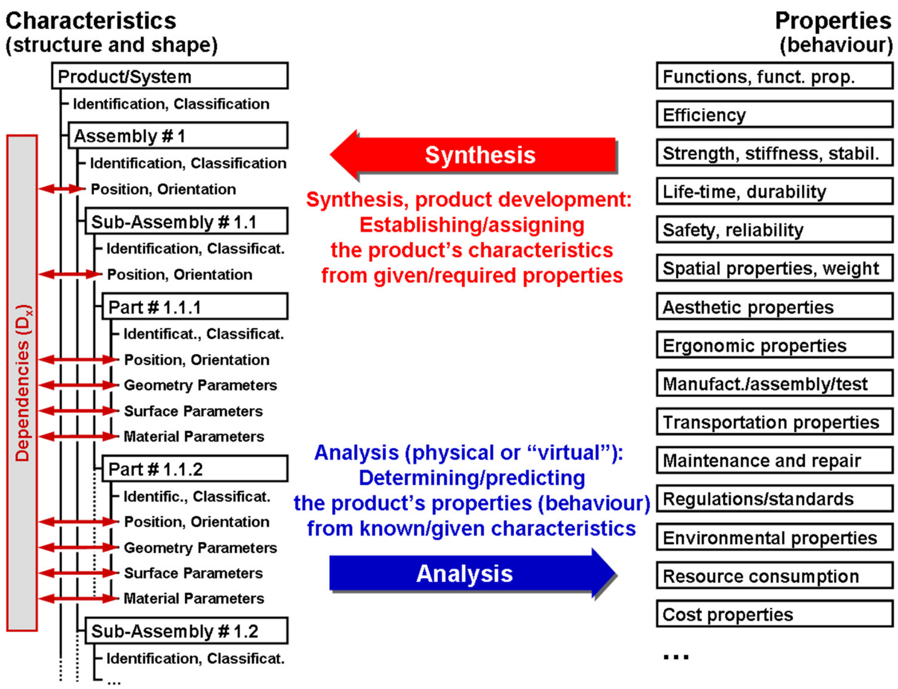

Figure 1. Characteristics and properties, and their two main relationships

On the characteristics (left) side of Figure 1, an additional block is drawn that represents dependencies (**D**$_x$) between characteristics. Designers are familiar with these types of dependencies, e.g. geometric or spatial dependencies, as well as those concerning fits, surface and material pairings, even conditions of existence. It should be noted that the existence of these dependencies is more an advantage than a complication for the design process: Each one of them reduces the number of characteristics (and, thus, the number of design degrees of freedom) by one – one less that the designer has to take care of.

Finally, Figure 1 shows the two main relationships between characteristics and properties:

- **Analysis:** Based on known/given characteristics (structural parameters, design parameters) of a product, its properties are determined (and therefore, its behaviour), or – if the product does not yet exist – predicted.
- **Synthesis:** Based on given, i.e. required properties (or in later stages: based on the gap between required and as-is properties), the product's characteristics are established and appropriate values assigned.

Synthesis is the main task of product development/design. The requirements list is, in principle, a list of required properties; the task of the development engineer/designer is to find appropriate solutions, i.e.

an appropriate set of characteristics that meet the requirements to the customer's or user's satisfaction. In many practical cases the requirements already contain certain characteristics which means that some partial solutions (solution patterns, see section 3) are set from the beginning.

Besides synthesis operations, designing also needs analysis operations: They serve the purpose of checking whether the as-is properties of the solution actually meet the required properties.

Using the symbols shown in **Table 1**, analysis and synthesis as the two main relationships between characteristics and properties can be modelled in more detail according to **Figure 2**. As a simplification, in these models both characteristics and properties are displayed as simple lists (no hierarchical or other structure). These lists of characteristics and properties, respectively, could also be noted as vectors $\underline{\mathbf{C}}$ and $\underline{\mathbf{P}}$ – similar to the approach proposed by [Suh 1990, 2001].

Table 1. Explanation of symbols used in the diagrams

$\mathbf{C_i}$	Characteristics (*Merkmale*)
$\mathbf{P_j}$	Properties (*Eigenschaften*)
$\mathbf{PR_j}$	Required properties
$\mathbf{D_x}$	Dependencies (constraints) between characteristics
$\mathbf{R_j}$	Relations between characteristics and properties – for analysis operations
$\mathbf{R_j^{-1}}$	"Inverse" relations betw. properties and characteristics – for synthesis operations
$\mathbf{EC_j}$	External conditions

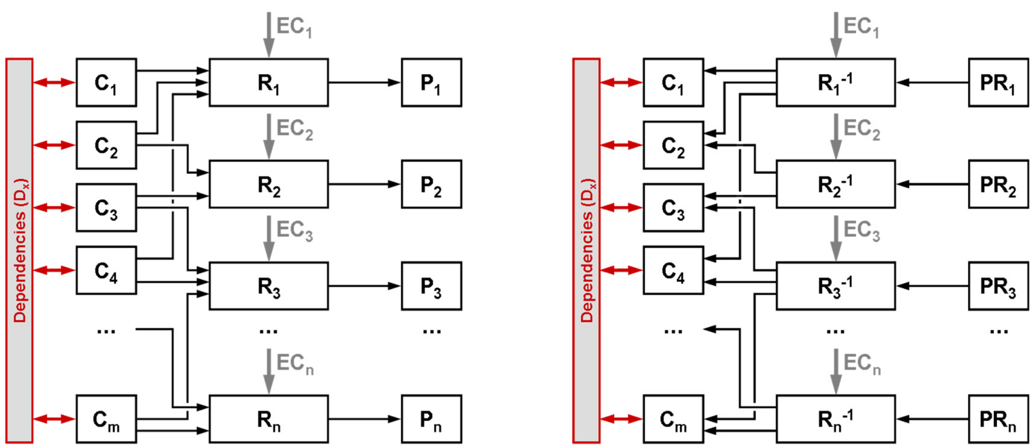

Figure 2. Basic models of analysis (left) and of synthesis (right)

The core content of the analysis model (Figure 2, left) is that for a product with given characteristics (analysis!) they determine all relevant properties; however, for each individual property a different combination of characteristics is constitutive.

Once the product exists (i.e. when the product's characteristics $\mathbf{C_i}$ are physically realised, "materialised") and operates, the analysis of its properties/behaviour ($\mathbf{P_j}$) can be performed by testing and measuring. In this case the product itself is the representation of the relations ($\mathbf{R_j}$). During product design, however, there is not yet a finished product: Its properties can only be analysed by means of appropriate methods and tools which are based on – physical or non-physical – models. The relation-boxes ($\mathbf{R_j}$) stand for these methods and tools; their purpose is to tell about the influences of relevant characteristics ($\mathbf{C_i}$) on the respective properties ($\mathbf{P_j}$), thus ***predicting*** the properties given at that moment.

The basic model of analysis (Figure 2, left) displays one more element: The determination/prediction of every product property via an appropriate model, method or tool must be performed with respect to certain external conditions ($\mathbf{EC_j}$). They define the framework in which the statement about the respective property is valid. Examples are: analysing the load capacity or the life-time of a design solution with respect to the external load conditions (and their distribution over time); statements on the manufacturability are always linked to the manufacturing system as an external condition; even assessing the aesthetic properties of a design may be dependent on the assumed cultural background of customers or users. Obviously, these external conditions are particularly important for "Design for X" (DfX) and correspondent DfX-strategies [Weber 2007].

Formally, the synthesis model emerges from the analysis model by inversion: Based on given properties – i.e. required properties (**PR$_j$**) – the characteristics of the solution are to be determined and values assigned. Therefore, the model of synthesis according to Figure 2, right, is the analysis model (left), now re-drawn with all arrows reversed. The relations between required properties (**PR$_j$**) and characteristics (**C$_i$**) now are denoted "inverse relations (**R_j^{-1}**): This follows common use of the term "inverse" in science (e.g. "solving inverse problems"), even though in design synthesis is the "normal" case.

It may be noted that already this very simple synthesis model displays the nature of design conflicts: Different required properties influence the same characteristic(s) and demand changes in different directions ($C_2 \ldots C_m$ in Figure 2, right). The classical example is maximising the stiffness – which requires the cross-section to be increased – against minimising the weight – which requires the cross-section to be reduced.

In the preceding paragraphs product-modelling has been addressed (Characteristics-Properties Modelling, CPM). From this a process model (Property-Driven Development/Design, PDD) can be developed: Product design is a process consisting of cycles that implies the following steps (Figure 3):

1. Synthesis: Starting from required properties (**PR$_j$**), characteristics (**C$_i$**) of the future solution are established. This is often achieved by adopting partial solutions from previous designs (solution patterns, see section 3). The synthesis step is always the first step in a design cycle – without defining or detailing at least some characteristics there is nothing to analyse and evaluate in the subsequent steps.
2. Analysis: In this step, the current properties (**P$_j$**, as-is properties) of the current state of the solution are analysed, based on the characteristics established so far.
3. Determining individual deviations: Next, the results of the analysis (as-is properties) are compared with the required properties. The deviations between the two (**ΔP$_j$**) are the still existing deficiencies of the current design.
4. Overall evaluation: The designer has to run an overall evaluation; extracting the main problems and deciding how to proceed, that is, pick the property/properties to be addressed next and select appropriate methods/tools for the subsequent synthesis-analysis-evaluation cycle.

From one cycle to the next, because of each synthesis step, more and more characteristics are established and their values assigned ("detailing"). The analysis steps of all cycles basically all deal with the same properties, but with a modified and/or extended set of characteristics, thus creating increasingly precise information about the product's properties/behaviour.

The main message of this approach is: The product development/design process as a whole is controlled or "driven" by evaluating the gap between required and as-is properties at the end of each cycle.

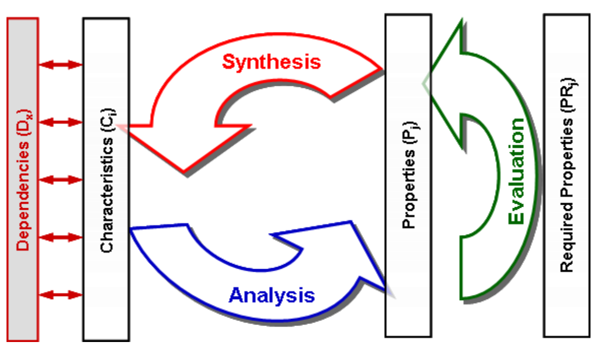

Figure 3. Scheme of the product development/design process consisting of cycles of synthesis–analysis-evaluation steps

3. Solution patterns (definition)

The CPM/PDD approach, as briefly described in section 2, delivers a good basis for the **definition of solution patterns** (or "solution elements"):

A solution pattern is an aggregation of characteristics (C_i) and properties (P_j) with known relations (R_j, R_j^{-1}) between the two, Figure 4. In coupling characteristics with properties, a solution pattern usually also implies – explicitly or implicitly – related external conditions (EC_j).
Together, characteristics, properties, the relations between the two and the related external conditions constitute the "knowledge" carried in the solution pattern.

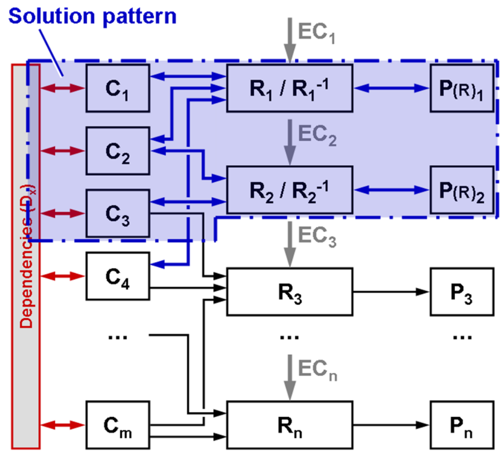

Figure 4. Solution pattern as an aggregation of characteristics (C_i) and properties (P_j) with known relations (R_j, R_j^{-1}) between the two

Solution patterns can be:

- Physical objects: Typical examples are machine elements, where we usually find the link between characteristics and properties as catalogues, tables, diagrams, calculation algorithms.
- "Virtual" objects: There is a lot of terms for digitised design patterns, e.g. variant programmes, pre-defined features and feature libraries, templates and – as a recent extension – "Knowledge-Based Engineering" (KBE).

It is very important to note that the term „solution pattern", as introduced here does not only contain functional/principle patterns (as is the case in traditional approaches to Design Theory and Methodology) but solution patterns that address **all** conceivable product properties. Some examples:

- Functional and principle patterns: Change speed and torque in a mechanical power transmission via gears, chain drives, belt drives, hydraulic or electric converters, …
- Strength and stiffness patterns: Certain cross-sections of components optimal for certain load types (e.g. T- or I-shaped sections for bending loads).
- Manufacturing patterns: Cast-iron vs. welded plates as concurrent patterns for housings; avoid castings in one-off manufacturing; … more or less all "Design for Manufacturing" (DfM) patterns and rules.
- Assembly patterns: Chamfers for easy assembly; modified screw heads (e.g. Torx) to ease automated tightening; … more or less all "Design for Assembly" (DfA) patterns and rules.
- Aesthetic patterns: Form-, colour-, touch-, sound-, odour-schemes signalling certain meanings.
- Use patterns: Side-stick control of an aircraft vs. the traditional control column ("yoke").

If characteristics (**C_i**), properties (**P_j**) and relations between them (**R_j**, **R_j^{-1}**) as constituents of solution patterns are all known, then this "knowledge" can be used in ***both*** directions:

- **Analysis:** Design patterns can help determining the properties based on given characteristics. This is often coupled with established, sometimes standardised analysis methods (**R_j**). Examples are the calculation of the strength of the solution pattern "gear pair" according to DIN 3990 or equivalent AGMA standards, or defined test routines for assessing the product's efficiency.
- **Synthesis:** Solution patterns allow reasoning from given required properties to the characteristics of the solution. The systematisation of solution patterns according to their properties is the basic concept underlying all design catalogues. While in the past design catalogues were situated in a particular domain (e.g. mechanical engineering [Roth 1982, 1994], [Koller and Kastrup

1998]), more recent publications extend the basic concept into a multi-domain perspective and investigate formalised description methods [Gausemeier et al. 2009], [Metzler et al. 2013].

Figure 5 uses two very simple examples to demonstrate: Solution patterns can be quite abstract and undetailed for early stages of the design process (Figure 5, left), but they can also provide "knowledge" for concrete detailing activities (Figure 5, right). Both examples shown in Figure 5 contain equations to illustrate analysis methods (**R**$_j$) that are usually integrated in solution patterns. These can be seen as elementary models supporting the next process step.

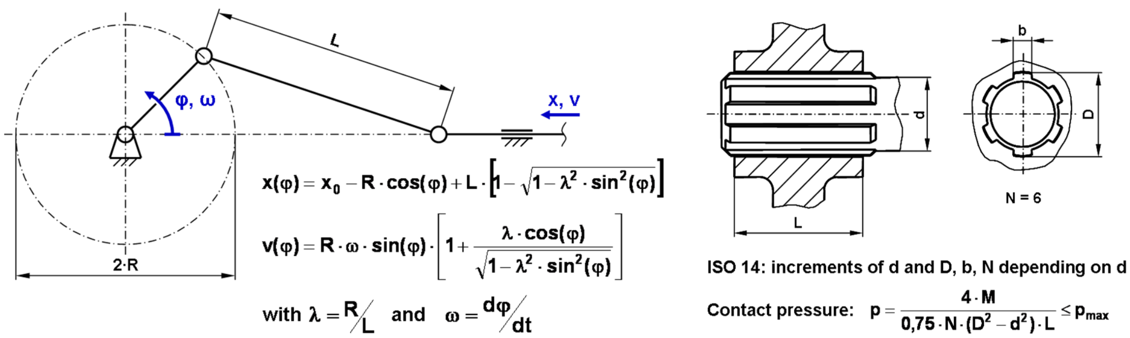

Figure 5. Examples of solution patterns for different stages of the design process. Left: Slider-crank mechanism as a potential solution principle for transforming a rotational into a linear motion (or vice versa) – still quite abstract and undetailed. Right: Defining details of forms and dimensions of a splined shaft-hub connection by standardisation [ISO 14]

In reality much larger "chunks" may be used as solution patterns, e.g. transferring the entire power train of a compact car from one product generation to the next. This concept links directly with considerations of [Albers et al. 2015] who put product **generation** development into focus as the standard case in industrial practice (opposed to new/original development/design that is the main concern of the traditional approaches to Design Theory and Methodology).

As an illustration, Table 2 shows an excerpt of solution patterns that are almost universally used in compact car design – some of them already for decades. For many other product areas – especially those that have a long history like motor cars – similar lists could be drawn up.

Table 2. Combination of solution patterns for the example of a compact car (excerpt)

Property	Established Solution Patterns
Drive	Primary drive by 3- or 4-cyl. internal combustion engine (Otto or Diesel, turbocharged) Power transmission via multi-gear transmission (manual shift or automatic) General layout front engine, transversally mounted, front-wheel drive
Suspension	Front: McPherson type, cyl. coil springs, telescopic hydraulic dampers, anti-roll bar Rear: Twist-beam type, cyl. coil springs, telescopic hydraulic dampers
Use	All use functions controlled by the driver … via steering wheel, traditional hand and foot controls
Aesthetics	Two-box shape, "hatchback" design
Manufacturing	Body-in-white constructed from pressed sheet steel, spot-welded Outer components (wings, bumpers) bolted; partly from plastic
Passive safety	Rigid passenger cell Deformation zones front/back ("crash-tubes/-boxes") Passenger restraint via 3-point seat-belts Front and side airbags
…	…

4. Solution patterns in product innovation

Following [Schumpeter 1934], “innovation” needs an “invention” as a pre-requisite plus its successful transfer into practical use. In the context of product design, we might even introduce another stage, i.e. “idea” as a pre-requisite of “invention” [Weber 2012]. Against this background it is debatable whether this section (and this contribution as a whole) deals with the role of solution patterns for the “idea stage”, the “invention stage” or the “innovation stage”. For reasons of simplicity, in the following considerations the most common (even if not entirely exact” term “innovation” is used.

The main message is:

Solution patterns can contribute to product innovation in two distinctly different ways:

A. Starting from an overall design solution consisting of an existing combination of solution patterns (as in many practical cases), replacing at least one of these solution patterns by an alternative one or adding a new solution pattern (“new” in the sense of “never used in this context”).
B. Creating a solution pattern that has not existed before (at least not in this composition) and developing it into a state fit for use in a subsequent design process.

Some examples and considerations may explain the differences between the two innovation types:

- Examples for innovations of the type A, here based on the compact car example (Table 2): The replacement of any of the solution patterns listed in Table 2 will cause considerable notion and will be seen as “product innovation”. Consider replacing the internal combustion engine as primary drive by an electrical drive or changing the steel body-in-white to an aluminium or fibre-reinforced plastic design, going from spot to laser welding, changing the controls to a joystick-device or providing autonomous driving functions, etc.
- Examples for innovations of the type B: Developing solution patterns for new functions of a product (e.g. autonomous driving), developing new manufacturing technologies (e.g. aluminium space-frame construction for car bodies, additive manufacturing for one-off or low-volume products), even developing new use techniques (e.g. using “wiping” gestures on touch-screens instead of keyboard interaction as an interface to IT devices).
- Innovation type A might be called “direct” product innovation, the type B “indirect”.
- Type A will in many cases utilise the outcome of innovations of type B.
- In type A the main concern is to ensure the compatibility of the solution pattern newly brought into the design with the other patterns (which might require modifying these as well).
- Innovation type B may require extensive basic research and proof of feasibility.
- The activities of most companies happen inside fairly stable application areas (like motor cars, Table 2) where specific combinations of solution patterns prevail. Innovations of the type A will hardly ever address the majority of these proven solution patterns; in most cases only one or two of them will be replaced by new approaches (“incremental innovation”).
- In these cases it will be necessary to adapt the neighbouring solution patterns, even if they look unchanged in the first place. Many errors can be done here, caused by ill-fitting of the “old” solution patterns to the newly introduced one(s).
- It may be noted that a new solution pattern as the outcome of an innovation type B can more or less directly be transformed into a patent: The property or properties addressed by that new solution pattern are the base of the patent claim, the characteristics describe the structure of the solution. [Koller 1998], chapter 12, makes interesting reading in this area if the definition of a solution pattern according to section 3 of this paper is added as background information.

Even if innovation of type A builds on type B results, both types are not necessarily performed by the same institutions:

- Doing innovation of the type A is the main task of companies (or of engineering service providers contracted by them). Public research institutions, e.g. university institutes, could do similar work (“applied research”), but normally confined to prototype studies.
- Innovation of the type B, being much closer to “basic research”, is quite often found in public research institutions, but also in technology start-ups.

- Some very successful large companies are known for their strategy of **not** going into type B innovation; instead, they systematically observe what is going on in this area and buy ideas or even companies who propagate them when results are promising (i.e. on the verge of being transformed into a type A innovation).

For innovation type A – the actual product innovation – the “goodness” of the new design solution heavily depends on the compatibility of the old and new solution patterns that are combined (maybe better: the relative minimum of conflicts between the solution patterns). The authors suspect that the reason for many innovation failures have been neglected conflicts between newly developed or applied solution patterns and the ones taken over from the past.

If the compatibility of solution patterns (or the relative absence of conflicts between them) is decisive for the success of a design, it is surprising that studies in this respect are quite rare in design research – despite the fact that major parts of design education are based on teaching solution patterns (see section 6). The authors of this contribution presume that recent works on complexity management (e.g. [Lindemann et al. 2009]) would be a good base for further investigations. In addition, the matrix method for change impact analysis presented in [Köhler et al. 2008] (which are also based on the CPM/PDD approach) could be checked for application.

5. Solution patterns in practical design

By observation of design processes in practice (however, in an unstructured way: no formal study has been performed) the authors venture to formulate the following **hypothesis**:

Hypothesis: The vast majority of practical design processes is executed by combining known, preferably well-proven solution patterns.

Design by “piling known solution patterns on top of each other” is, of course, an efficient way of re-using knowledge, reducing design effort and minimising risk.

However, many of the solution patterns, while addressing different properties, will contain the same characteristics. An example based on Table 2: The thickness(es) of the sheet steel for the body-in-white of the car influence(s) at the same time the ease of manufacturing (forming, welding), the passive safety, the weight, subsequently the performance (acceleration), the exterior and interior noise, etc. Independence of these solution patterns can rarely be achieved, contrary to the independence axiom propagated by [Suh 1990, 2001].

It should be noted again that the compatibility of the solution patterns (the relative absence of conflicts) that are combined is decisive for the “goodness” and the success of an overall design solution; more considerations on this issue were already described at the end of section 4. Obviously in certain application areas (like motor cars, see Table 2) specific combinations – which we might call “meta-patterns” – have proven particularly successful.

6. Solution patterns in design education

Based on the definition of solution patterns as presented in section 3, one can see that major parts of design education consist of explaining existing patterns:

- Teaching machine elements is an important part of the early design education (at least in continental Europe): Machine elements are well-known solution patterns which usually do not only address functions (transfer, guide, scale up/down, transform, merge/split forces and motions) but also strength, manufacturing and other properties.
- Courses on DfX (mainly Design for Manufacturing/Assembly, DfM/DfA) in engineering curricula usually consist of showing a wealth of solution patterns in the respective areas.
- Courses on Design Methodology are often based upon the “traditional European schools”, going back to [Hansen 1955], extended and more broadly published by [Pahl and Beitz 1977/2007] and [VDI-Guideline 2221, 1986/1993], both internationally best known via the English versions [Pahl and Beitz 1983/2007] and [VDI-Guideline 2221, 1987], respectively.

  When these “schools” were first published, the new thing was that in addition to object-oriented solutions patterns (like, for instance: machine elements) they introduced more abstract patterns based on functions and solution principles. A lot of effort went into catalogues collecting and presenting these new-type patterns [Koller 1976, 1998], [Roth 1982, 1994].

It may be noted that teaching design via solution patterns is not confined to mechanical engineering:

- Basic courses in electrical engineering present electrical elements, basic network patterns, solution patterns for amplifiers, converters, comparators, etc.
- Basic courses in computer science typically consist of showing proven data-structures, algorithms, interfaces and programme architectures for certain applications.
- As already mentioned, recent publications increasingly address domain-spanning solution patterns, their formal description and (re-) use [Gausemeier et al. 2009], [Metzler et al. 2013].

Again it is surprising that we teach different types of solution patterns in many places of design education – but independent of each other. We expect that students solve the problems of superimposing these different types of solution patterns (i.e. ensuring their compatibility, recognising conflicts, etc.) themselves. The authors wonder whether more focus on integration and compatibility of solution patterns would improve the outcome of design education. However, as was stated already in section 4, we may have little to offer as there is no research in this respect upon which we could build.

Engineering education is still mainly organised according to the classical disciplines (mechanical, electrical/electronic, information processing, service engineering). However, commercially relevant products increasingly consist of combinations of solution patterns out of these disciplines (e.g. "mechatronic" systems, Product-Service Systems). Therefore interesting questions for design education are:

- Which solution patterns to teach?
- Maintain discipline-specific teaching or make a new, inter-disciplinary approach?
- If yes, how to come to integrated explanatory concepts for the solution patterns taught?

Already in 1997 a group of German university professors coming from engineering design raised these questions and provided initial answers ("Heiligenberg Manifesto", [Albers and Birkhofer 1997]), this initiative renewed in 2013 under the auspices of Member of WiGeP (*Wissenschaftliche Gesellschaft für Produktentwicklung*, Academic Society for Product Development, [WiGeP 2013]).

7. Conclusion and further steps

In this contribution a definition of the term "solution pattern" is presented and, reasoning from there, the role of solution patterns in design innovation is studied, with additional views on design practice and education. Main messages:

- Most design processes in practice consist of combining known, preferably well-proven solution patterns.
- Product innovation can be divided into two cases: (A) replacing one or a few solution patterns in a design or (B) developing an entirely new solution pattern.
- Solution patterns play an important role in teaching design.

In all cases, the compatibility (maybe better: the relative minimum of conflicts between the solution patterns) of the solution patterns combined in an overall solution plays an important role. However, this issue is not very widely addressed in design research. It is suggested to invest more effort in this direction for which the transfer of findings from work on complexity management [Lindemann et al. 2009] and/or the matrix method for change impact analysis [Köhler et al. 2008] are seen as potential starting points.

References

Albers, A., Birkhofer, H., "Die Zukunft der Maschinenelemente-Lehre – Heiligenberger Manifest, Vorträge, Workshopergebnisse", Schloss Heiligenberg, 1997.

Albers, A., Bursac, N., Wintergerst, E., "Product Generation Development – Importance and Challenges from a Design Research Perspective", Mastorakis, N.E., Solomon To, C.W. (eds.), Proceedings of the International Conference on Mechanical Engineering (ME 2015), TU Wien, Austria, 2015, pp. 16-21.

Gausemeier, J., Frank, U., Donoth, J., Kahl, S., "Specification Technique for the Description of Self-Optimizing Mechatronic Systems", Research in Engineering Design, Vol.20, 2009, pp. 201-223.

Hansen, F., "Konstruktionssystematik – eine Arbeitsweise für fortschrittliche Konstrukteure", VEB Verlag Technik, Berlin. 2nd ed., Available online, <www.tu-ilmenau.de/konstruktionstechnik/>, 1955.

Hubka, V., Eder, W. E., "Design Science", Springer, London, 1996.

Köhler, C., Conrad, J., Wanke, S., Weber, C., "A Matrix Representation of the CPM/PDD Approach as a Means for Change Impact Analysis", Marjanovic, D., Storga, M., Pavkovic, N., Bojcetic, N. (eds.), Proceedings of Design 2008 (DS 48), Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Vol.1, 2008, pp. 167-174.

Koller, R., "Konstruktionsmethode für den Maschinen-, Geräte- und Apparatebau", Springer, Berlin-Heidelberg. 1st ed. 1976, 4th ed., 1998.

Koller, R., Kastrup, N., "Prinziplösungen zur Konstruktion technischer Produkte", Springer, Berlin-Heidelberg, 1998.

Lindemann, U., Maurer, M., Braun, T., "Structural Complexity Management – an Approach for the Field of Product Design", Springer, Berlin Heidelberg, 2009.

Maier, J. R. A., Fadel, G. M., "Affordance: the Fundamental Concept in Engineering Design", Proceedings of ASME Design Theory and Methodology Conference, Pittsburgh, PA, 2001.

Maier, J. R. A., Fadel, G. M., "Affordance Based Design: a Relational Theory for Design", Research in Engineering Design 20, 2009, pp. 13–27.

Metzler, T., Mosch, M., Lindemann, U., "Meta-Design Catalogs for Cognitive Products", Chakrabarti, A., Prakash, R. V. (eds.), Proceedings of the International Conference on Research into Design ICoRD'13, “Global Product development”, Springer, New Dehli, 07.-09.01.2013, 2013, pp. 303-316.

Pahl, G., Beitz, W., "Engineering Design", Springer, London, 1st ed. 1983, 3rd ed. 2007.

Pahl, G., Beitz, W., "Konstruktionslehre", Springer, Berlin-Heidelberg. 1st ed. 1977, 7th ed. (by Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H.), 2007.

Roth, K., "Konstruieren mit Konstruktionskatalogen", Springer, Berlin-Heidelberg. 1st ed. 1982, 2nd ed. (in two volumes) 1994.

Schumpeter, J., "The Theory of Economic Development", Harvard University Press, Cambridge, MA, USA, 1934.

Suh, N. P., "The Principles of Design", Oxford University Press, 1990.

Suh, N. P., "Axiomatic Design", Oxford University Press, 2001.

VDI-Guideline 2221, "Systematic Approach to the Design of Technical Systems and Products", VDI, Düsseldorf, 1987, English version of VDI-Guideline 2221, 1986.

VDI-Guideline 2221, "Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte", VDI, Düsseldorf, 1986. 1st ed. 1986, updated version 1993.

Weber, C., "Idea – Invention – Innovation. Strategies, Approaches, Research Challenges", Marjanovic, D., Storga, M., Pavkovic, N., Bojcetic, N. (Eds.), Proceedings of Design 2012 (DS 70), Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, 2012, pp. 1265-1274.

Weber, C., "Looking at ‘DFX’ and ‘Product Maturity’ from the Perspective of a New Approach to Modelling Product and Product Development Processes", Krause, F.-L. (ed.), Proceedings of the 17th CIRP Design Conference, "The Future of Product Development", Springer, Berlin-Heidelberg, 2007, pp. 85-104.

Weber, C., "Modelling Products and Product Development Based on Characteristics and Properties", Chakrabarti, A., Blessing, L.T.M. (eds.),"An Anthology of Theories and Models of Design", Springer, London, 2014, pp. 327-352.

Wissenschaftliche Gesellschaft für Produktentwicklung (WiGeP), "Leitfaden universitäre Lehre in der Produktentwicklung", WiGeP 26.09.2013. Available online: <http://www.wigep.de/fileadmin/download/wigep/WiGeP_Leitfaden_Lehre_V01-20130926__final.pdf>, 2013.

Christian Weber, Univ. Prof. Dr.-Ing.
Technische Universität Ilmenau, Engineering Design
Max-Planck-Ring 12, D-98693 Ilmenau, Germany
Email: christian.weber@tu-ilmenau.de