

WHAT IS DESIGN KNOWLEDGE FROM THE VIEWPOINT OF CPM/PDD?

J.Conrad, C. Köhler, S. Wanke and C. Weber

Keywords: design knowledge, CPM/PDD, product knowledge, process knowledge

1. Introduction

Knowledge is a very common term in engineering design and also extensively discussed. There are many theories describing what knowledge is and how it should be managed in engineering design. But no shown/presented concept or definition is generally accepted. One reason for this are differing theories and views about the underlying development/design process. Therefore, it is necessary to find an adequate model for knowledge, which considers the product development process.

This paper is based on the Characteristics-Properties Modelling / Property-Driven Development (CPM/PDD) approach [Weber 2005]. The product model (i.e. CPM) and the model of the product development process (i.e. PDD) are based on the clear distinction between characteristics and properties and, thus, enable a structured and integrated consideration of design knowledge. On this account, this paper provides a closer look on design knowledge in the context of CPM/PDD. Among other considerations, a method out of the theoretical computer science is applied to this context.

2. Background and related work

2.1 Data, information and knowledge

In theoretical considerations, knowledge is often described by the differentiation in data, information and knowledge [Mertens et al. 2005]. The correlation is expressed as an enrichment process in three steps: the lowest step is data, then information as middle step and knowledge as the highest one (figure 1).

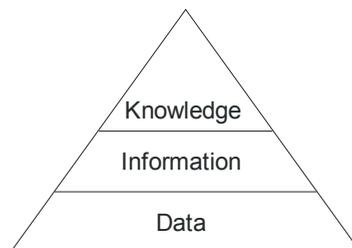


Figure 1. Data, information, knowledge [Mertens et al. 2005]

This strict hierarchical distinction is very common in literature but also quite vulnerable. In the opinion of the authors, the clear distinction, especially between information and knowledge, leads to

more confusion than to added value. A more open representation is proposed by Probst [Probst et al. 2006]. He describes the issue as a continuum of data, information and knowledge (figure 2).

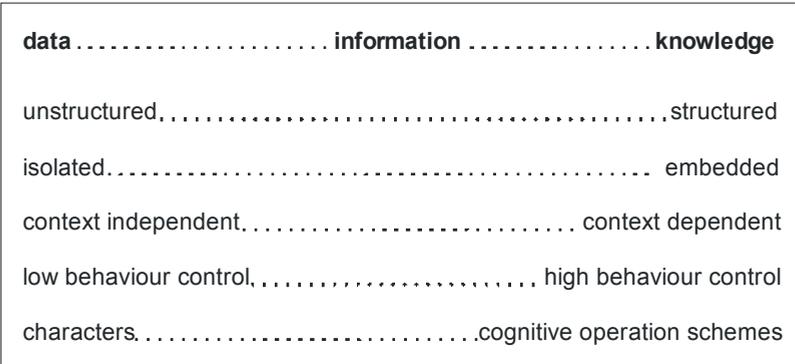


Figure 2. Data information knowledge continuum (following [Probst et al. 2006])

The pairs of terms (unstructured-structured, isolated-positioned, context independent-context dependent, etc.) define indicators for the distinction between data, information and knowledge. But due to the non-discrete character of a continuum, a clear distinction is not possible and even not intended; the transitions are fluent.

Hicks [Hicks et al. 2002] offers a more differentiated distinction of information and knowledge. These considerations define classes of formal and informal informational information and knowledge elements and knowledge processes.

Another issue often discussed in literature [Probst et al. 2006] is the distinction between explicit and implicit knowledge. The following does not focus on this distinction; nevertheless, it should be considered more deeply when analysing possible data-sources for information and knowledge.

This paper follows the representation shown in figure 2 as it represents the interconnection of data, information and knowledge differentiated enough but not too complicated and detailed for the context of this work.

2.2 Design Knowledge

Many approaches exist to describe design knowledge. According to Muth [Muth 1994] and Weber [Weber et al. 1991], design knowledge consists of two parts: product or object knowledge on the one hand and process knowledge on the other hand. Thereby product knowledge is the knowledge about the technical part or system, which is designed. Process knowledge contains the knowledge about how the designer fulfils requirements on a product, which has to be developed. This implies knowledge about every step like imaginations in the designers mind, the detailing and finally the real product realisation.

Also Vajna (for instance [Vajna 2001]), Meerkamm (for instance [Meerkamm et al. 1991]), Hicks [Hicks et al. 2002] and others developed interesting and valuable approaches to describe design knowledge. But the approach of Weber seems goal-oriented and promising for the considerations of this paper and will be basis in the following.

2.3 Characteristics-Properties Modelling / Property-Driven Development (CPM/PDD)

This section explains briefly the Characteristics-Properties Modelling / Property-Driven Development (CPM/PDD) approach. Prof. Christian Weber developed the CPM/PDD approach at Saarland University [Weber 2005]. The concept of CPM represents the product model, PDD depicts the process of developing and designing products based on CPM.

CPM/PDD is based on the distinction between characteristics and properties of a product:

- The characteristics (C) describe “the structure, shape and material consistency of a product (“*Struktur und Gestalt*”, “*Beschaffenheit*”)” [Weber 2005]. A very important aspect of them is that they can be directly determined and influenced by the designer (e.g. geometry etc.).
- The properties (P) describe “the product’s behaviour” [Weber 2005]. They can not be directly determined and influenced by the designer (e.g. weight, safety and reliability, aesthetic properties), only indirectly by modifying the characteristics.

The links between characteristics and properties are represented by relations. Two types of relations can be distinguished which correspond with the two main activities in the product development process: In *analysis*, characteristics are known and – via analysis relations (R) – the product’s properties are derived. In *synthesis*, properties are known/required and – via synthesis/“inverted” relations (R^{-1}) – the product’s characteristics are established. The characteristics, properties and the two relation types are shown in figure 3.

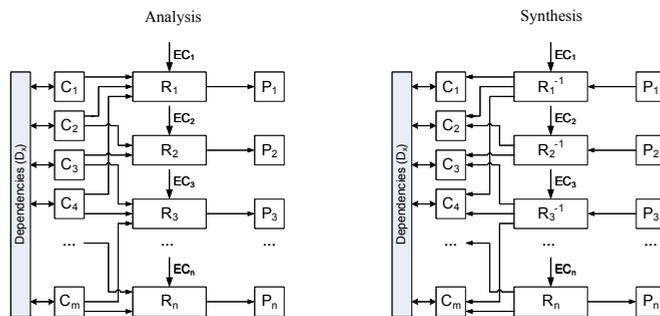


Figure 3. Analysis and Synthesis in the CPM Model [Weber 2005]

To respect constraints on the characteristic side, this model contains dependencies between them (D). Finally, external conditions (EC) represent the context in which (analytical as well as synthetical) statements are valid.

PDD describes the product development process based on the CPM product model. In principle, a given set of required properties (Soll-properties) should be met by the behaviour of the design solution (Ist-properties) as good as possible. In a new product development project, (almost) nothing is known about the structure and characteristics of the solution at the beginning. In case of an adaptive or variational design, a list of characteristics is “set” from the beginning, together with the properties they bring along.

The PDD process consists of several cycles, each one of those having 4 steps (shown in figure 4):

- Synthesis step: Determine and assign characteristics from the overall evaluation (in the first step of new product development only from the Soll-properties)
- Analysis step: Determines or predicts Ist-properties from the given characteristics
- Individual deviation: Evaluates every Ist-property against its Soll-property
- Overall evaluation: Draws conclusions from the individual deviations and drives and controls the process, i.e. strategy for synthesis or termination of the process. The overall evaluation is the driver of the PDD process

If new properties appear by adding characteristics and relations, the designer has to decide whether they are relevant or not, i.e. whether they should be added to the set of properties.

The four steps of the PDD process are passed in several cycles, until all relevant Ist-properties fulfil the Soll-properties sufficiently. Thereby, the Ist-properties have to be determinable and predictable with sufficient certainty and accuracy. Additionally, for termination it is necessary that all characteristics needed for manufacturing and assembly of the product are established and assigned.

In the following, the external conditions as well as the dependencies are not considered any further to enable basic considerations without getting lost in details.

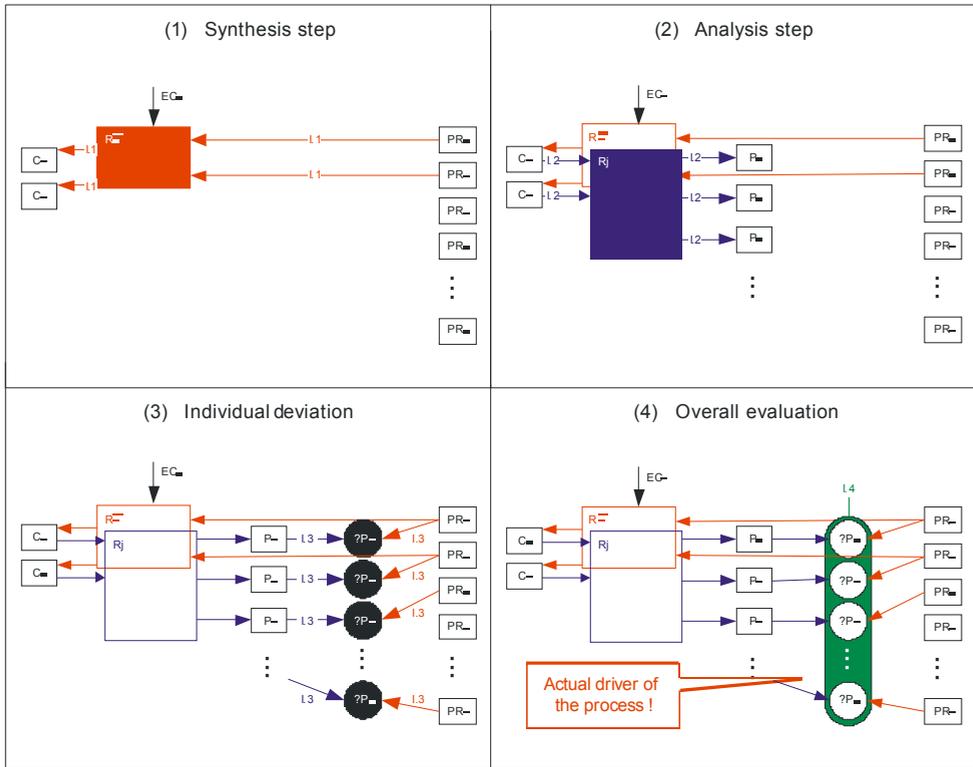


Figure 4. PDD process steps [Weber 2005]

3. Design knowledge on the basis of CPM/PDD

As written in section 2.2., design knowledge is fairly often divided into product and process knowledge. The following section 3.1. discusses the product knowledge side on the basis of CPM/PDD. The conclusions of this section are basis for the considerations on process knowledge, taken at section 3.2.

3.1 Product knowledge on the basis of CPM

The product model in CPM consists of characteristics, properties and relations (other elements not discussed here). The list and values of the characteristics of a product are determined by the designer and describe the products structure and shape. These are for instance documented in technical drawings, CAD files etc. The property list contains information about the solution's behaviour and the required properties. These could be requirements lists, documentations, test reports, result of analyses (for instance FEM) etc. This information about characteristics and properties in CPM can be considered as product information according to the data-information-knowledge continuum (figure 2). The information about characteristics and properties can be located in the middle. They are more than just characters like data. But they are rather isolated and not very context depending.

Out of the viewpoint of product development, the connection of characteristics and properties through relations can be considered as product knowledge [Wanke et al. 2007]. Relations represent the context of the characteristics and properties and show, if and how they are linked. They show, how changes on

characteristics affect the properties side. The relations in the CPM model can be formulae and tables, but also simulation tools, physical mock-ups and expert knowledge.

In the data-information-knowledge continuum (figure 2), it is obvious that the higher degree of interconnectivity, context dependency and structure leads to a shift from information to knowledge. So, the relations in the PDD model can be considered as the driver that turns the product information into product knowledge.

Thus, information about all characteristics, properties and their relations is needed, in order to speak about product knowledge in the context of CPM (figure 5).

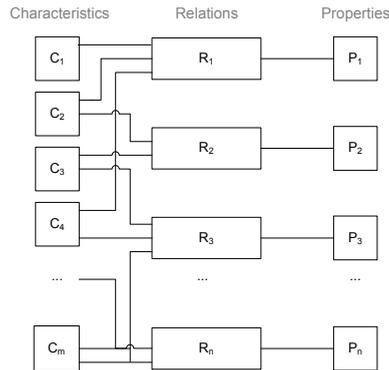


Figure 5. Product knowledge

3.2 Process knowledge on the basis of CPM/PDD

“The technical knowledge about designing in its highest form is process knowledge”[Hubka et al. 1996]. By this quotation the importance but also the complicated character of process knowledge is evident. The interpretation of process knowledge in CPM/PDD is more complicated than the description of product knowledge. A good method in mathematics and computer science is to transfer unknown problems to known issues. This method is introduced shortly in the following; the aim is to reduce the complicated issue of process knowledge to the easier interpretable issue of product knowledge.

3.2.1 Reduction in computer science

In theoretical computer science, especially in the computability theory and for decision problems, the concept of reducibility is very common. It is motivated by the following consideration:

If it is possible to *reduce* the problem P1 to a special case of the unknown problem P2, so it is possible to draw conclusions about P2 out of the knowledge of P1 [Schöning 2001][Sipser 2005]. Thereby, the reduction has to be done in decision problems by a total and computable function [Schöning 2001]. The most fundamental reducibility notion is the Turing reducibility, which enables considerations whether a problem is Turing computable or not [Sipser 2005]. These questions are important in order to decide which problems are describable in algorithms and which are not. This background seems to be very theoretical, but it plays a significant role in compiler design and in the building of programming languages.

3.2.2 Reduction of process knowledge to product knowledge

Out of the previous section, the following considerations try to transfer this technique and reduce the process knowledge question to the better-known problem of the product knowledge. In the framework of the CPM/PDD approach explained in section 2.3, it is necessary to reduce the PDD process to the CPM model. In this chapter, the term “solution” refers to the product solution, which is to be developed by passing the PDD process.

According to section 3.2.1., it is necessary to find a function, which maps the PDD process to a CPM model. A CPM model consists of characteristics, relations and properties, hence, the identification of the corresponding parts of the PDD process is required. Following chapter 2.3., characteristics are directly determinable and influenceable by the designer. This means that they are the claviature the designer can play in order to change the product's behaviour. In the PDD process, the designer can directly add characteristics to the developing solution, change them or drop them to affect the process. So, these operations could be considered as the "characteristics" of the PDD process. "Properties", i.e. the "behaviour" of the process, are the result of the evaluation of the evolving solution's overall behaviour. This result determines the next step, whether characteristics of the process should be changed (and which one) or whether the process is finished. A premise of this consideration is an objective overall evaluation and not a blandishing to influence the process. The relation between properties and characteristics in CPM describes the way how characteristics and properties are related. The characteristics and properties discussed above are related by the CPM model of the currently evolving solution. So this CPM model is the "relation" of the transferred PDD process. These considerations allow a reduction of the PDD process to the CPM model. The relation is computable because it is an easy mapping and total because it fits for every design process in PDD. Figure 6 shows a product development process with the PDD approach, expressed by a CPM model via the developed reduction. This model describes exactly one step of the PDD model.

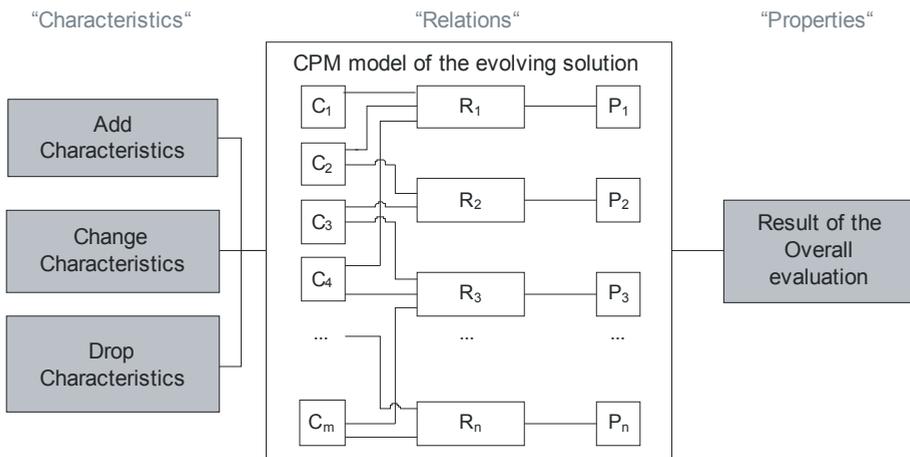


Figure 6. PDD process as CPM model

3.2.3 Process knowledge

If we assume that the knowledge of the product is characterised like described in chapter 3.1., it is possible to derive out of the reduction shown in figure 6 what process knowledge in PDD is. Information about the characteristics, properties and the relation between them (see figure 6) have to be available.

- "Characteristics" of the process are the possibilities to add, change and drop characteristics in the current solution.
- "Properties" of the model above are the result of the overall evaluation-step in PDD. To do an overall evaluation, information about Ist-properties, Soll-properties and the ability to evaluate them is necessary. To obtain a result out of the evaluation, the ability to draw conclusions is crucial. This conclusion is the strategy of the next cycle: adding, changing, dropping characteristics or finishing the development process.

- A “Relation” is the CPM model of the evolving solution. Here, the product knowledge of the evolving solution like described in chapter 3.1. is necessary

Process knowledge is, according to section 2.2., the knowledge about “how” to drive the process. Recapitulatory, the presented reduction shows that in PDD the following aspects are in each step of the product development process crucial:

- product knowledge of the evolving solution
- ability to evaluate Ist-properties against Soll-properties of the evolving solution
- ability to draw conclusions out of the evaluation

The tricky part of the process knowledge and also of the design knowledge at all is the non-deterministic character of the design process. The strategy for the next cycle in the process (i.e. the result of the overall evaluation) of the product development process has to adapt dynamically to the process, depending on the current evaluation and situation.

4. Discussion and further work

Knowledge is a very important and significant topic in engineering design. Even if it is not an explicit matter, it plays implicitly a role in the background. CPM/PDD is one possibility to illustrate the product development process. Therefore, a closer consideration of knowledge in CPM/PDD can be useful.

Design knowledge is dividable in product knowledge and process knowledge. This approach is used in the presented consideration. The description of product knowledge in CPM/PDD is thereby straightforward. The more complicated part is the description of process knowledge. Therefore, the method of reduction is transferred from theoretical computer science into the context of engineering design. The chosen reduction of process knowledge (knowledge in PDD) to product knowledge (knowledge in CPM) gives a hint to the possible interpretation of process knowledge. Together with the interpretation of product knowledge, the result indicates what design knowledge is, especially in CPM/PDD.

The presented approach is not intended to be proof like the reduction in computer science. It should give a basis or hints for this highly complicated issue. In future work, the results of this work can be justified in uses cases and compared with other design knowledge approaches like [Vajna 2001], [Meerkamm et al. 1991] and [Hicks et al. 2002]. These considerations could be basis for further work on managing knowledge and knowledge sources, in particular in product development with CPM/PDD. In addition, the impact of dependencies and external conditions is a possible subject for further considerations.

References

- Hicks, B.J., Culley, S.J., Allen R.D., Mullineux G., “A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design”, *International Journal of Information Management* Vol 22, pp 263-280, 2002.
- Hubka, V., Eder, W. E., „Design Science“, Springer Verlag, Berlin, Heidelberg, 1996.
- Meerkamm, H., Weber, A., “Konstruktionssystem mfk Integration von Bauteilsynthese und Analyse“, *VDI Berichte Nr.903, Erfolgreiche Anwendung wissensbasierter Systeme in Entwicklung und Konstruktion*, Springer Verlag, Berlin, Heidelberg, 1991, pp. 231-248.
- Mertens, P., Bodendorf, F., König, W., “Grundzüge der Wirtschaftsinformatik“, Springer Verlag, Berlin, Heidelberg, 2005.
- Muth, M., „Repäsentation von Konstruktionswissen unter Verwendung des objektorientierten Paradigmas“, *Schriftenreihe Produktionstechnik, Band 7, Universität des Saarlandes, Saarbrücken*, 1994
- Probst, G., Raub, S., Romhardt, K., “Wissen managen Wie Unternehmen ihre wertvollste Ressource optimal nutzen“, Gabler, Wiesbaden, 2006.
- Schöning, U., „Theoretische Informatik - kurzgefasst“, *Spektrum Akademischer Verlag, Heidelberg, Berlin*, 2001.
- Sipser, M., “Introduction to the Theory of Computing“, *PWS Publ. Comp.(Int. Thomson Publ.)*, Boston, 2005.

Vajna, S., "Wissensmanagement in der Produktentwicklung", 12. Symposium "DfX", Neukirchen 11.-12.10.2001, Band 12, 2001.

Wanke, S., Conrad, J.; Köhler, C., „Verhaltensbeschreibende Produktkataloge - Ein Anwendungsbeispiel der Solution Patterns des CPM/PDD Ansatzes“, 18. Symposium "DfX", Neukirchen 11.-12.10.2007, Band 18, 2007.

Weber, C., "CPM/PDD – An Extended Theoretical Approach to Modelling Products and Product Development Processes", 2. German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes, TU Berlin/Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik (IPK), Bley, H.; Jansen, H.; Krause, F.-L.; Shpitalni, M. (Ed.), Fraunhofer-IRB-Verlag, Stuttgart, 2005, pp. 159-179.

Weber, C., Stark, R., "Wissensbasierte Systeme für die Konstruktion – Grundlagen aus Konstruktionsmethodischer Sicht", 8th International Conference on Engineering Design 1991 (ICED 91), Zürich/Schweiz 27.-29.08.1991. Schriftenreihe „Workshop–Design–Konstruktion“, WDK 20: Proceedings of ICED 91 (herausgegeben von V. Hubka), Vol. 2, S. 1151-1162. Heurista-Verlag, Zürich 1991.

Dipl.-Inform., M.Sc. Jan Conrad
Institute of Engineering Design/CAD, Saarland University
PO Box 15 11 50, 66041 Saarbrücken, Germany
Tel.: +49 / (0) 681 / 302-58304
Fax.: +49 / (0) 681 / 302-4858
Email: conrad@cad.uni-saarland.de
URL: <http://www.cad.uni-saarland.de>