NordDesign 2008
August 21 – 23, 2008
Tallinn, Estonia

A Functional Model for the Function Oriented Description of Customer-Related Functions of High-Variant Products

Daniel P. Politze
Function Engineering Methods
Daimler AG
Hanns Klemm Str. 45
71034 Böblingen
GERMANY
daniel.politze@daimler.com

Stefan Dierssen
Autonomous Systems Lab
Swiss Federal Institute of Technology
Tannenstr. 3
8092 Zürich
SWITZERLAND
dierssen@mavt.ethz.ch

Abstract
In automotive industry the number of product functions realized by software or electronics increases every year. As a consequence complexity grows and functions cannot be clearly assigned to a single component any more. Function orientation and thus the function oriented description of product functions along the development process is seen as a promising approach to master the complexity of today‘s products.

Experiences at Daimler have shown that users face difficulties when building up a function oriented product description (FOPD). A reason for that is that the description becomes very extensive and confronts a user with high complexity.

In this article we present a functional modeling approach for representing a FOPD, suitable for high complex and high variant products. Based on such a model, professional tool development can be started, which is still missing.

Keywords: Functional Modeling, Function Oriented Product Descriptions, Customer-related Functions, Automotive Industry

1 Introduction
The growth of system complexity in automotive industry is mainly driven by the fact that product functions are more and more realized by the combination of mechanical, electric/electronic, and software components.

We assume that customers want functions and they think in functions as depicted in figure 1 (left hand side), because when a customer has a problem with his car he normally complains about defect functionality and not about certain components (figure 1, right hand side). In the same way he demands reliable functions, when he is looking for a new car.

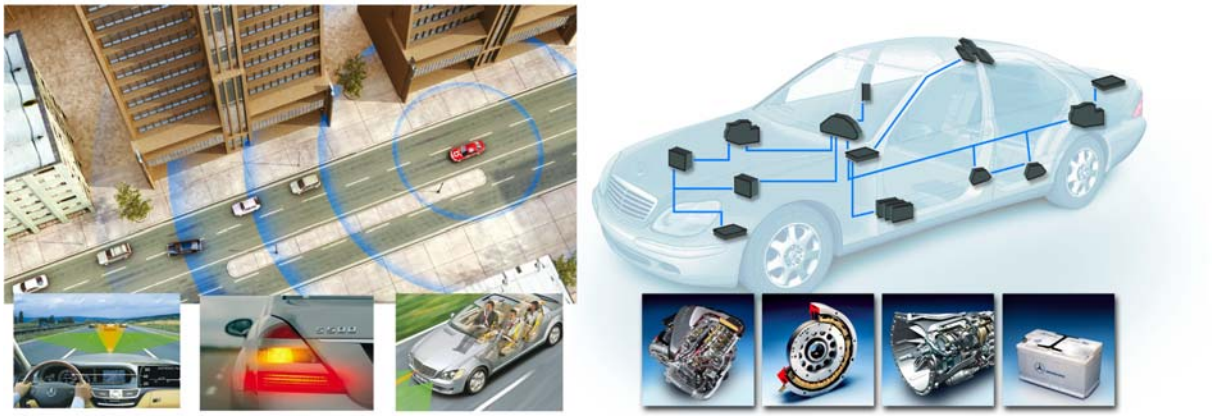

Figure 1. Changing paradigm: thinking in functions, not in components

Therefore, we infer that the quality of a product is also perceived by its functionality. That means, in order to provide high quality product functions, it is not sufficient to bring mechanical, electric/electronic and software components to perfection separately. Hence, the component‘s interactions and their contributions to product functions become important and should be considered very early in the design process. For this reason product functions shall be captured and described, resulting in an explicit function oriented product description (FOPD), which allows reuse and improvement of existing function specifications and thus evolves to a mature function oriented product specification.

The creation of a FOPD is not an easy task. Especially for high variant products, the description becomes very extensive and confronts a user with high complexity. This is mainly driven by the high number of product functions, the increasing share of software and the fact that the description of a function may differ between product variants. In this context, tool support for creating, managing and using a FOPD is demanded, which in turn requires an appropriate representation in form of a functional model.

Such a functional modeling approach is described within this paper which is structured as follows: In the next section related literature is reviewed. In section 3 the shortcomings of existing functional modeling approaches are determined followed by the description of required modeling demands in section 4. Based on that a new model approach is proposed in section 5 followed by a short example on how to use this model in section 6. Finally section 7 concludes and opens up future research perspectives.

2 Related Work

In the past decades, functions and the modeling of functions have become a part of some well-known design methodologies such as of Roth [21], Pahl and Beitz [18], Hubka and Eder [10], Ulrich and Eppinger [27], Suh [23, 24] or Ullman [26]. In order to support a common understanding of functions between all stakeholders in the design process formal function representations and vocabularies have been defined and most of them assume an abstract verb-object representation of functions, as influenced by earlier work on Value Engineering [17, 28]

Collins et al. [5] presented a very specific vocabulary that contained the description of 105 functions of a mechanical system in the helicopter domain. Similar, but not restricted to a certain domain, are the descriptions of basic functions provided by Altshuller [2]. Based upon a very large empirical study he states the theory of inventive problem solving and lists 30 physical effects and phenomena that may be used for the solution of design tasks. Later, Malmqvist et al. [16] compared this theory with the systematic approach of Pahl and Beitz [18] and found out that the differences between the corresponding function vocabularies are due to the level of abstraction and not to the content.

Collins et al. [5] and Altshuller [2] did not explicitly distinguish functions and their associated flows, which Pahl and Beitz [18] did. Thus, Pahl and Beitz provided five general function classes, which are based on the findings of Krumhauer [13], and in addition to that they differentiate between material, energy and signal flows. This vocabulary has been refined by Hundal [11], who proposed six basic functions and four very abstract item categories. Motivated by the National Institute of Standards and Technology (NIST), Szykman et al. [25] developed and presented a generic classification scheme, containing about 130 types of engineering functions and 100 types of flows. Independently of the NIST effort, Stone et al. extended the previous research of Little et al. [14] and came up with a functional vocabulary with 32 functions and 28 flows, what is known as the functional basis for design [22]. In this work the authors also compared existing vocabularies and argued that the approaches of Pahl and Beitz [18], Hundal [11] and Altshuller [2] are subsumed by the functional basis.

Because of a high degree of similarity, the two rivaling vocabularies, defined by Szykman et al. [25] and Stone et al. [22] were reconciled by the work of Hirtz et al. [8], which integrated the differences and derived a more comprehensive and evolved functional basis for design.

Not all approaches are referring to a verb-object representation of functions. Based on the work of Collins et al. [5] a Function-Based Taxonomy has been proposed by Kirschman and Fadel [12]. Comparable to a verb-adjective representation they provide four basic function groups, which may be refined with connotations according to their taxonomies, resulting in a more detailed function description. Even more detailed function descriptions were the result of Eisenhut [6] and his pragmatic functions. He allows a less abstract and more domain specific verb-object representation, that may be enhanced by subjects, adjectives and adverbs.

In the last decade there has also been some work on functional models that takes aspects from the environment into accout. In [4] a functional model is presented that sees a device introduced in a certain world and interacting with it in terms of structural relations or actions. This work is mainly based on the idea that someone is doing something or has a property that is desired by someone else. Different viewpoints were distinguished and integrated in this work, which led to the conclusion that a device may play several different roles with respect to its environment and may provide a desired function, when used as intended. A very similar approach, that also integrates the point of view of a particular actor is given by Maier and Fadel [15], where the potential uses of a device are called affordances. A comparison of these two approaches may be found in [3].

3 Problem statement

As stated in the introduction, it is important to build up a holistic function oriented product description (FOPD) which evolves to a mature functional product specification. Especially in the automotive industry this is seen as a key factor for mature products with high quality functionalities. Many people have argued that there is more to engineering design than technical problem solving and this viewpoint is also reinforced by recent literature. Especially work in the field of user-centered design that originates from the software domain shows that understanding and integrating users in the design process points in the right direction. Subsequently the work of Houdek [9] and Heumesser et al. [7] describe the need and challenges for mature product specifications and raise the question for an adequate description approach. Based on that Allmann [1] suggests an additional abstraction layer for specifying customer-related product functions, which can be compared to the level of abstraction of a user manual. Independently, Ringler et al. [20] presents a comprehensive development approach for E/E-architectures that uses descriptions of customer-related product functions as input.

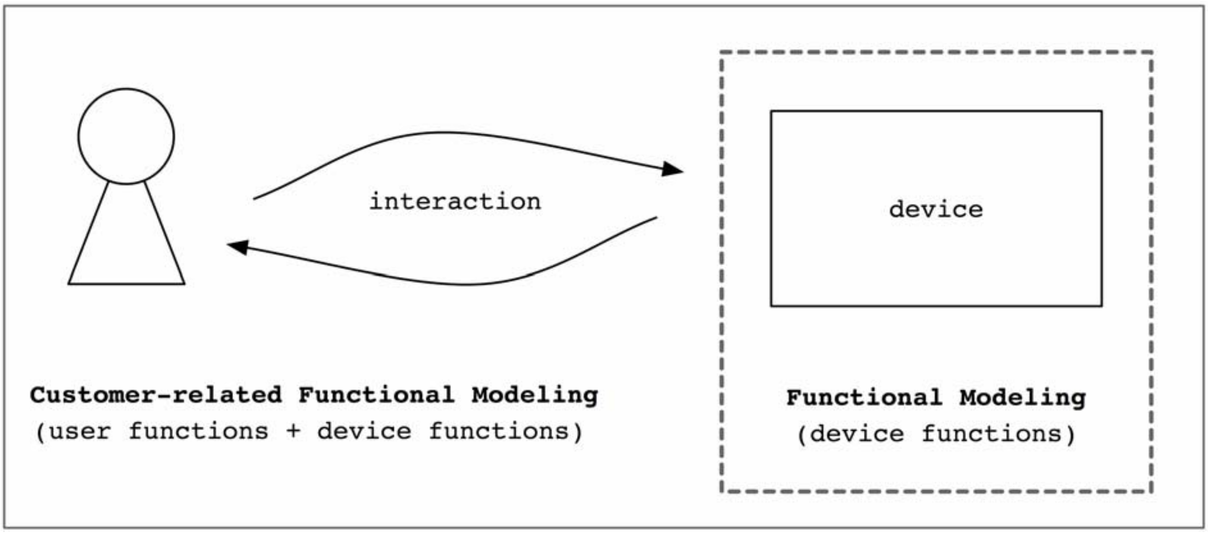

Figure 2. The scope of customer-related functional modeling

Given these findings on what is required it is easy to recognize that a formal model for representing customer-related functions is needed. Unfortunately most functional modeling approaches (see section 2) focus on technical aspects and although some include different viewpoints they are not sufficient for describing customer-related functions of high complex mechatronic products in a formal way. In addition to that, existing approaches usually aim at the descriptions of device function and not user functions, as differentiated in [8]. This implies that necessary modeling demands for the description of customer-related functions are missing. Thus a formal model is needed, that is appropriate for customer-related functional modeling (see figure 2), where a device may also be seen as a black box.

4 Modeling demands for customer-related functional modeling

As we have learned from the experience at Daimler, additional modeling demands are required for the proper description of customer-related product functions. In the following the most important demands are explained in detail.

Demand for „Activities“

Let us assume the following situation: It‘s raining and the windshield wiper is activated. The driver is about to pick up a friend. Whenever his friend arrives at the car, he will open the door and the wiper will stop in the same moment to prevent wet clothes.

In this case a customer-related function „prevent wet clothes“ can be identified, which may further be described as „open door“ stops „move wiper“. Thus there is a need to express some sort of activities between functions. That means in some cases it is necessary to say that a function „starts“, „pauses“ or „stops“ another function.

Demand for „Sequences“
Based on the description of activities, another problem may be stated. Therefore one should think on an automated hatch, imagine the hatch is lowered automatically and something is in its way. Of course one wants the hatch to stop and to return to its initial position.

The customer-related function may be named as „prevent clamping at hatch“. This function could then be described as „detect clamping“ stops „close hatch“ and then starts „open hatch“. That means the activities between functions have to described again and additionally a chronological order has to be specified. Hence, there is need to define sequences between activities. This can be simply done by describing that a function starts another function „before“ or „after“ starting another function as it is depicted in figure 3.

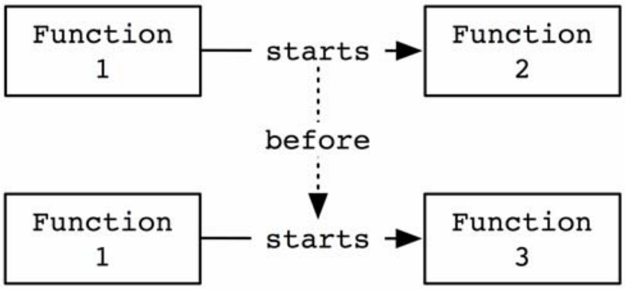

Figure 3. Activities and sequences in a FOPD

Demand for „Preconditions“
For the explanation of this problem we refer to the hatch example again and assume there is a button holding the function „start opening hatch“ in the interior, which causes „open hatch“ to start. It is easy to understand that a customer does not want a hatch to open while driving, even though he might have accidently pressed the button.

To express that it should be possible to describe „start opening hatch by button“ starts „open hatch“ only if „vehicle is not moving“. This implies, that we need a way to define preconditions for the availability functions as well as for the activities between them.

Demand for „Variability“
Especially Daimler has to deal with high complex products, such as a Mercedes S-Class with a multitude of product variants. This in turn has a big impact on the function descriptions due to the fact that the functions and their realization may differ between product variants, e.g. “open hatch” may be achieved either manually or as in the example above automatically after pressing a button. Hence variability information has to be captured.

Basically there exist two fundamental ways how to do that. The first method is to keep the variability information inside a single FOPD and the other possibility is to have a FOPD for each product variant.

As pointed out in [19] it is advantageous to have the variability information integrated in a FOPD in order to supports reuse and transparency aiming at a mature functional product specification. As a consequence information about product variability must also be considered in a formal model for representing customer-related functions.

5 Functional Model Approach

In this section we present a formal function model that is appropriate for describing customer-related functions of high complex mechatronic products. The model presented here allows traditional functional modeling and provides a solution for the modeling demands stated in the previous section.

Because a customer-related function may be realized by a composition of subfunctions, functional decomposition is also part of our model, like in most functional models. Furthermore we follow the paradigm of Pahl and Beitz where a hierarchy of functions is working on flows. Based on that we assume a function to have inputs and outputs and define them as explicit parts of our model. In this way we extend the traditional distinction between function and flow by describing their connection with corresponding input descriptors and output descriptors.

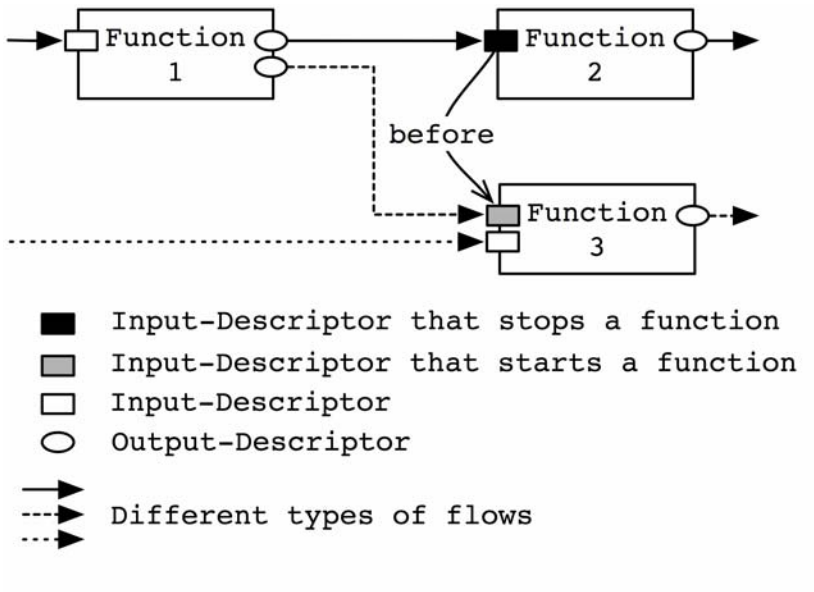

Figure 4. The idea of having functions, flows and descriptors

Figure 4 shows the basic idea of our functional model, with different types of flows and functions depicted as large white boxes. For every outgoing flow, there exists an output descriptor, represented by a small white ellipse. Similar to that there is an input descriptor for every incoming flow. Depending on the type of associated activity, an input descriptor is either represented as a small coloured box. In this example white and black are used, which means a flow starts or stops a function respectively. Additionally there is a chronological order defined between two input descriptors. Thus, from the above figure we can infer that „function 1" first stops „function 2" and then starts „function 3".

At this point it should be clear how „Activities" and „Sequences" may be represented, which addresses half of the modeling problems from section 4. The solution for „Preconditions" and „Variability" requires special attributes on functions and descriptors.

The preconditions and variability information are simply added by the of use a character string. Strings have been successfully applied for this purpose in bill of materials (BOM) systems in the automotive industry. Also strings allow the formulation of variability information and preconditions in natural language and therefore ensures enough expressiveness.

In figure 5 the models for the representation of functions, flows and descriptors are given. Each has an identifier and its definition is surrounded by braces.

Representation of functions

According to the given definition every function must have a single „name“. We encourage to use a very pragmatic solution as suggested in the work of Eisenhut [6]. In order to be consistent with existing functional modeling approaches, the „type“ of a function may be specified according to a function class taxonomy, such as the functional basis [8]. „var“ and „cond“ are the attributes for providing information about varibility and preconditions. „sub“ is to refer to other functions in the sense of functional decomposition. These functions are considered as subfunctions. „in“ and „out“ are references to descriptors that are intended to describe the input and the output of a function. It is easy to see, that our model of a function is conform with the idea of having at least one input and one output.

```
Function{                                  Descriptor{
   name    String         1                   type    DescriptorType   1
   type    FunctionType   1                   flow    Flow             1
   var     String         0..1                var     String           0..1
   cond    String         0..*                cond    String           0..*
   sub     Function       0..*                act     ActivityType     0..1
   in      Descriptor     1..*                before  Descriptor       0..1
   out     Descriptor     1..*                after   Descriptor       0..1
   props   Property       0..*             }
}
                                           Flow{
                                              name    String      1
                                              type    FlowType    1
                                           }
```

Figure 5. Model schemes for function, flow and descriptor

Representation of descriptors

Since descriptors are very unique there is no need to name them. However it is required to specify its „type“, which may be either input descriptor or output descriptor. Also necessary is the reference to a single „flow“, the descriptor is meant for. The next two items are „var“ and „cond“, which are already known from the scheme for functions and serve the same purpose, which is to provide information about variability and preconditions.

The remaining items are „act“, „before“ and „after“, which are intended for describing activities and sequences. Those have to be specified for input descriptors only. act is for specifying the type of activity that happens when the referenced flow is detected. From our current experience we propose „starts“, „stops“, „pauses“ and „resumes“ as adequate types of activities. before and after point to other input descriptors and thereby define a chronological order between them.

Representation of flows

As in the case of functions, each flow must have a unique „name“. One reason for that is that it might help to check consistency. Another reason is that it allows to express particular events. Such events can be interpreted as binary information signals and labelled with pragmatic names e.g. „button pressed“. The last item of the flow scheme is meant for the

„type“ of flow. For the same reasons as they apply for functions this type should be described according to an existing flow taxonomy.

6 Example

In this section the usage of the functional model proposed in this article shall be demonstrated. Therefore a modeling of the wiper-example from section 4 is presented in figure 6 and explained in detail.

In this case we want to model the customer-related function „prevent wet clothes when opening door“. This function is available in two products X and Y, which is expressed by the var attribute. It can be inferred from figure 6 that the function has three subfunctions, that are „open door by handle“, „move wiper over windshield“ and „indicate wiper hold“.

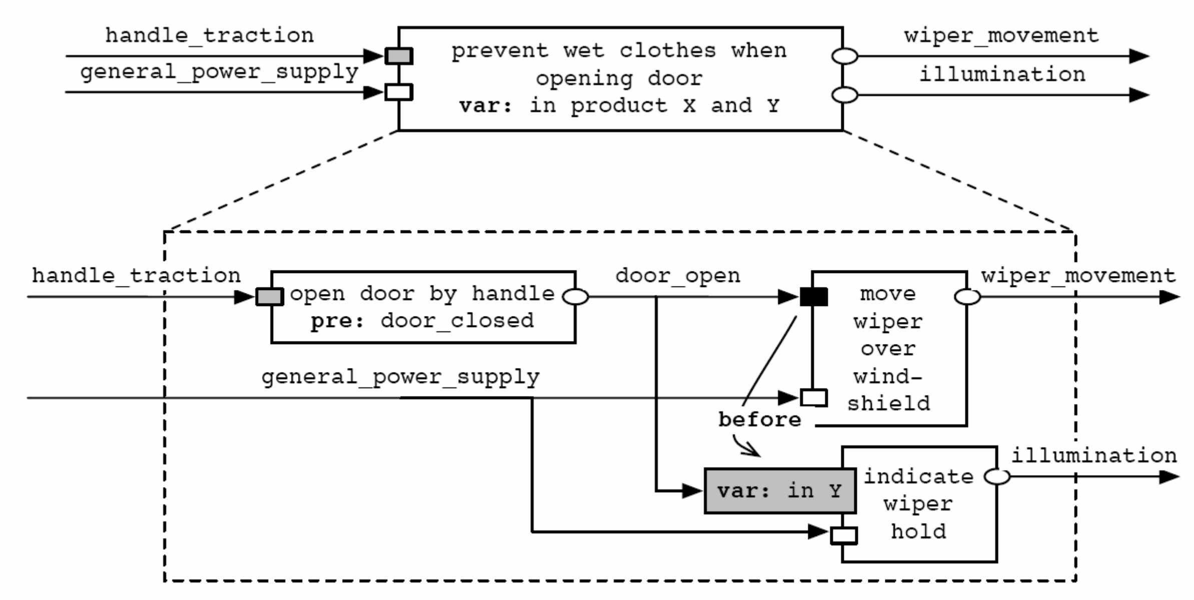

Figure 6. Modeling example

Provided that the door is closed, „open door by handle“ is started by an energy flow, named „handle_traction“, which automatically serves as input. The output of this function is a signal flow that indicates „door_open“. Of course it should also cause the door to open, but this is omitted to maintain clarity. „move wiper over windshield“ transforms electrical power to a force labelled as „wiper_movement“ and may be stopped by the signal „door_open“. The last subfunction „indicate wiper hold“ is intended to give a feedback to the driver. Therefore a lamp is assumed that transforms „general_power_supply“ into „illumination“ whenever „door_open“ is detected.

It is also depicted in figure 6 that the „door_open“ signal causes „move wiper over windshield to stop“ before „indicate wiper hold is started“. Furthermore „indicate wiper hold“ is only started in product Y as expressed by the „var“ attribute.

7 Conclusion

This article shows that there is an enormous growth in complexity in today‘s automobiles and that mastering this complexity is seen as a key to success for the future. In the context of providing high quality product functions to the customers, an explicit and holistic function oriented product description (FOPD) becomes important and that customer-related functions define an appropriate level of abstraction.

An extensive literature study showed that none of the existing functional modeling approaches are suitable for customer-related function modeling, since they are primarily focusing on device functions. We further identified four modeling problems, which have to be solved in order to allow customer-related product functions and thus build up a FOPD.

Finally a functional modeling approach has been presented that bases on the separation of functions, flows and descriptors for the input and output of a function. Even though it has been shown that this model is already very expressive, we see further modeling demands, such as states and state transitions that will be needed soon.

However, this work has shown that there is reason to shift functional modeling to a more customer-related view and that this shift raises new research opportunities.

Currently we are looking on the development of tools that allow easy editing and managing of function descriptions based on the proposed model. Future work will be necessary on support for consistency checks as well as interfaces to existing functional vocabularies.

References

[1] C. Allmann. Anforderungen auf Kundenfunktionsebene in der Automobilindustrie. In SE 2007 – die Konferenz rund um Softwaretechnik, Hamburg, März 2007.

[2] G. S. Altshuller. Creativity as an Exact Science. Gordon and Branch Publishers, 1984.

[3] D. C. Brown and L. Blessing. The relationship between function and a ordance. In Proceedings of the2005 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, number DETC2005-85017, Long Beach, California, USA, September 2005.

[4] B. Chandrasekaran and J. R. Josephson. Function in device representation. Engineering with Computers,16:162–177, 2000.

[5] J. Collins, B. Hagan, and H. Bratt. The failure-experience matrix - a useful design tool. Journal of Engineering in Industry, 98:1074–1079, 1976.

[6] A. Eisenhut. Service Driven Design: Konzepte und Hilfsmittel zur informationstechnischen Kopplung von Service und Entwicklung auf der Basis moderner Kommunikations-Technologien. PhD thesis, Eidgenössische Technische Hochschule Zürich, 1999.

[7] N. Heumesser and F. Houdek. Towards systematic recycling of systems requirements. In Proceedings of 25th International Conference on Software Engineering, 2003. pages 512–519, 2003.

[8] J. Hirtz, R. Stone, D. McAdams, S. Szykman, and K. Wood. A functional basis for engineering design: Reconciling and evolving previous efforts. Research in Engineering Design, 13(2):65–82, 2002.

[9] F. Houdek. Requirements Engineering Erfahrungen in Projekten der Automobilindustrie. Softwaretechnik-Trends, 23(1), 2003.

[10] V. Hubka and W. E. Eder. Theory of Technical Systems: A Total Concept Theory for Engineering Design. Springer Verlag, Berlin and New York, 1988.

[11] M. S. Hundal. A Systematic Method for Developing Function Structures, Solutions and Concept Variants. Mechanism and Machine Theory, 25(3):243–256, 1990.

[12] C. Kirschman and G. Fadel. Classifying functions for mechanical design. Journal of Mechanical Design, Transactions of the ASME, 120(3):475–482, 1998.

[13] P. Krumhauer. Rechnerunterstützung für die Konzeptphase der Konstruktion. PhD thesis, TU Berlin, 1974.
[14] A. D. Little, K. L. Wood, and D. A. McAdams. Functional analysis: A fundamental empirical study for reverse engineering, benchmarking and redesign. In Proceedings of the 1997 ASME Design Engineering Technical Conference and Computers in Engineering Conference, DETC97/DTM-3879, Sacramento, CA, September 1997. American Society of Mechanical Engineers.
[15] J. R. A. Maier and G. M. Fadel. Affordance-Based Methods for Design, In Proceedings of the 2003 ASME Design Theory and Methodology Conference, DETC-2003, Chicago, USA.
[16] J. Malmqvist, R. Axelsson, and M. Johansson. Comparative analysis of the theory of inventive problem solving and the systematic approach of pahl and beitz. In Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference, 96-DETC/DTM-1529, Irvine, CA, August 1996. American Society of Mechanical Engineers.
[17] L. Miles. Techniques of Value Analysis Engineering. McGraw-Hill, 1972.
[18] G. Pahl and W. Beitz. Konstruktionslehre: Methoden und Anwendung. Springer, 7th edition, 2007.
[19] D. P. Politze. Experiences on Including Variability Information in a Function Oriented Product Documentation. In J. Gausemeier, F. J. Rammig, and W. Schäfer, editors, Self-optimizing Mechatronic Systems: Design the Future, HNI-Verlagsschriftenreihe, Paderborn, pages 53–63. Heinz Nixdorf Institut, Heinz Nixdorf Institut, Feb. 2008.
[20] T. Ringler, M. Simons, and R. Beck. Reifegradsteigerung durch methodischen Architekturentwurf mit dem E/E-Konzeptwerkzeug. In 13. Internationaler Kongress Elektronik im Kraftfahrzeug, 10./11. Oktober 2007, Baden-Baden, Düsseldorf, 2007. VDI Verlag.
[21] K. Roth. Konstruieren mit Konstruktionskatalogen. Springer, Berlin, 1982.
[22] R. Stone and K. Wood. Development of a Functional Basis for Design. Journal of Mechanical Design, 122(4):359–370, 2000.
[23] N. P. Suh. The principles of design. Oxford University Press, New York, 1990.
[24] N. P. Suh. Axiomatic Design Theory for Systems. Research in Engineering Design, 10(4):189–209, 1998.
[25] S. Szykman, J. W. Racs, and R. D. Sriram. The representation of function in computer-based design. In Proceedings of the 1999 ASME Design Engineering Technical Conferences, DETC99/DTM-8742, Las Vegas, NV, September 1999.
[26] D. Ullman. The Mechanical Design Process. McGraw-Hill, New York, 1997.
[27] K. T. Ulrich and S. D. Eppinger. Product Design and Development. McGraw-Hill, New York, 1995.
[28] Value Analysis Incorporated. Value Analysis, Value Engineering and Value Management, Clifton Park, new York, 1993.