INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

DESIGN 2008

KNOWLEDGE ORIENTED PROCESS MANAGEMENT FOR DFX

M. Faerber, F. Jochaud, C. Stöber, S. Jablonski and H. Meerkamm

Keywords: process management, ontology, DfX

1. Introduction

Nowadays process development reaches much farther than just fulfilling requested product features. Instead increasing requirements from customers, the market or even law make product development more and more complex. In order to cope with these, product developers need support, which can be found in the in the "Design for X" (DfX) approach. DfX can be understood as rules or best practices on how certain requirements can be satisfied while developing a product. It provides a rule or knowledge system where best practices about the proper implementation of technical specifications are explicitly specified. Besides the direct requirements of specifications, all the implications from other aspects of the development process must be taken into account. They have to be considered and implemented during early stages of the product development process to avoid rework or double work in the later phases. As many decisions must be made at the beginning of a product development, all implications of a decision should be known by product developers. This help in decision making is provided in the DfX approach.
In order to provide the right information to product developers, the FORFLOW research cooperation is developing a process navigator. The process navigator is meant to be the central information system to coordinate product development processes (PDP) and provide information for developers. It is built on top of a workflow management system and displays information on the current one as well as it proposes the next possible steps. The aim of the process navigator is to provide both flexibility and support during product development processes.
In order to ensure that the process navigator can be adapted to new requirements and fit into the new context, its architectural design uses a model driven, process based approach. A model driven approach means that logic and content of the process navigator are not hard coded and can be exchanged. It is then possible to easily adapt the process navigator to new requirements. The following two different models are used:

- *Process Model*. The process model describes the development process itself, the dependencies between work steps and documents produced. A detailed explanation of its contents is provided in Section 2.1.
- *Knowledge Model*. The knowledge model describes the context and best practices that can be used while working on a certain process step. A major part of this model is the DfX-model which will be detailed in Section 2.2.

By combining these two models an integrated process oriented view on product development is achieved and developers can be supported effectively. As these two models are originally independent from each other, we will show how they can be integrated into a single model so that DfX criteria can be evaluated for process steps. After describing the concept, its implementation in a software system using ontologies and description logics will be shown. In the last section we will demonstrate how this model is used in the process navigator and how the contained information is presented to developers.

2. Models

In this section, we present our process modelling approach based on a simple example taken out of a PDP and the models of DfX criteria and methods structuring. Based on these two models, we then propose a solution to integrate them for their use in the Process Navigator.

2.1 Process Model

The perspective oriented process modelling approach [Jablonski96] is used to model development processes. One of the key features of this approach is that its extensibility allows all requirements from companies to be included in the process model. The process modelling tool (i>PM) that implements this model is using a meta-data repository to achieve this flexibility. The perspective oriented process modelling approach is composed of several so called perspectives. Each perspective describes the process under a different viewpoint and adds different information to the process model. The combination of all perspectives constitutes an integrated view on the process model. The following five perspectives usually form the basis for a comprehensive process model.

- *Functional perspective*: The functional perspective describes the task of the process step. Each process step can be decomposed into several subtasks.
- *Data perspective*: The data perspective describes which data (or documents) a process step consumes or produces. Thus the input and output data of a process step is defined. All inputs and outputs together build up the data flow in the process model.
- *Behavioural perspective*: The behavioural perspective describes the order in which process steps have to be executed.
- *Organisational perspective*: The organisational perspective defines persons or roles that are responsible for the execution of a given task.
- *Operational perspective*: The operational perspective defines tools or systems that support the execution of a process step.

Although the different aspects can be regarded as separated models, only their combination establishes a complete process model. With these five perspectives the basic question: "*Who* is performing *which task* using a specific *tool* to create or produce a certain *product (document)*" can be answered.

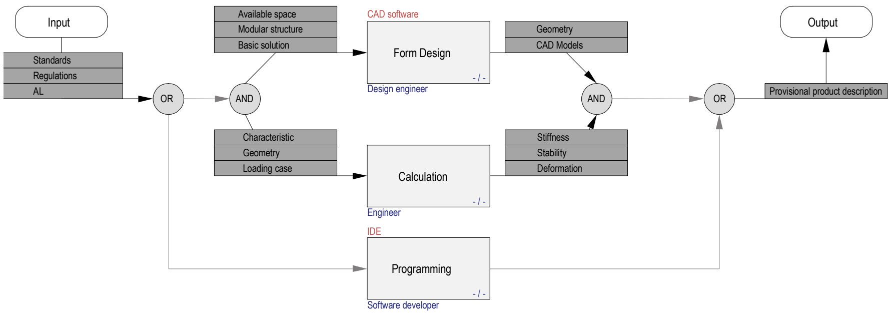


Figure 1. Process model

In Figure 1 an example process model is shown. It is taken out of a PDP and describes the detailed construction of a component. The work steps (functional perspective) are show in big rectangles with tools (operational perspective) and roles (organizational perspective) described on the top and bottom. They are connected by data flow (data perspective) or control flow (behavioural perspective).

2.2 DfX model

In order to make DfX knowledge available to product developers in the process navigator, a suitable connection of the DfX methods with the DfX approach is created, which must be fitted into the PDP.
In the literature no consistent structuring approach of DfX can be found. Two prevailing structuring approaches of DfX exist: the hierarchical approach of BAUER [Bauer03] and the structuring of the DfX criteria corresponding to their origin in the specific product life phases according to RUDE [Rude98]. Based on these two, a structuring approach for process methods was developed, that fosters tracing DfX criteria to their origins on the one hand; on the other hand the new approach provides developers with suitable methods depending on the quality of the available context information and data in the PDP.
The advantage of the hierarchical approach of BAUER is that it emphasizes the interrelation between the phases in the product development and the hierarchy of the DfX guidelines. The gradual implementation of the product in the PDP is related to the structuring of the DfX guidelines, which are highly affected by the character of the specific development phases. Regarding the structuring of the DfX Criteria according to RUDE, it becomes clear that a multitude of influencing factors have to be taken into account while developing and constructing a product designed for life cycle. This approach improves the understanding of the different requirements, by tracing the specific style and DfX guidelines to their origins in the product life cycle.
However in this approach, no exact allocation between the DfX guidelines and phases of the PDP takes place. The model we developed combines the benefits of the two mentioned structuring approaches. Our aim was to create a basic structure emphasis the connections of the most important influences from the product life cycle to the corresponding phases of the PDP. This way, it facilitates the allocation of DfX criteria and methods to the specific phases of the PDP, in which they are used.
The figure 2 depicts our approach to structure DfX criteria from their most general influences to their subdivision in basics instructions. Note that the relations we use between the levels of the DfX model and the Method model are defined for the figures 3, 4 and 5 as follows:

- *X is kind of Y*: defines that X is a specialized form of a more general concept Y.
- *X is a Y*: defines that X is an instance of a specific class Y.
- *X is part of Y*: defines that Y can be decomposed in X plus other concepts.

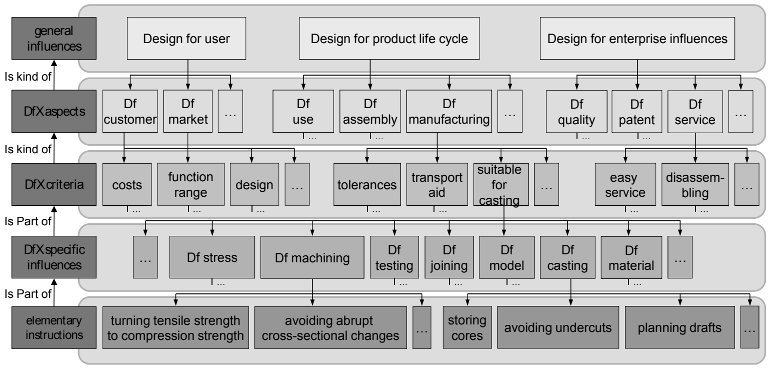


Figure 2. Structuring levels of the DfX approach

In the developed model (fig. 2) the following levels are depicted:

- *General influences*: These influences generally affect a product. They are derived from the user, from the product life cycle and from internal and external company influences, like standards, laws, used technologies, enterprise strategies and general competence of the company.
- *DfX Aspects*: From this group DfX aspects like designing for customer, market, manufacturing, recycling, service or quality can be derived. For tracing processes, knowing the origin of corresponding requirements and DfX criteria is important.

- *DfX Criteria*: DfX criteria are used to retrace the origin and importance of requirements to either the customer or the company during the PDP. While progressing in the PDP, engineers convert more and more requirements to concrete properties of a product, observing the decisions of earlier phases.
- *DfX Specific Influences*: They are a decomposition of the DfX criteria and are used to add additional information to them. For instance, while designing components suitable for casting, different DfX specific influences must be considered like designing for machining, load or model.
- *Elementary Instructions*: Basic instructions give concrete suggestion how certain DfX criteria have to be implemented in a process step. The engineer will be supported by basic instructions constructing products suitable for casting like “mounting cores“, “avoiding undercut“ or “turning tensile strength to compression strength“. Several instructions can be found in the corresponding literature.

The implementation of the *DfX criteria*, which are detailed by *DfX Specific Influences*, is made by appropriate *Elementary Instructions*. Both are displayed in the process navigator and made available to engineers. Methods can be described as a procedure building on a rule system, which serves for the acquisition of scientific findings or practical results, in order to support a developer in his activities. If a DfX criterion is implemented with a certain method, then this method is called *DfX method*. The aim is to prepare and connect existing methods to DfX criteria so that they can be used in the process navigator. Based on the already existing method structuring (see [Roth94] [VDI2221]) they were arranged according to the objective function, since this is purpose-leading for the coupling of the DfX approach with the methods and the integration into the PDP. The methods are divided according to the recurrent activities, which an engineer has to perform during the PDP [Wallmeier01], for example "analysing", "searching solutions" or "evaluating". Different method levels exist as per fig. 3, whereby specificity of the methods increases more and more on the progressive levels.

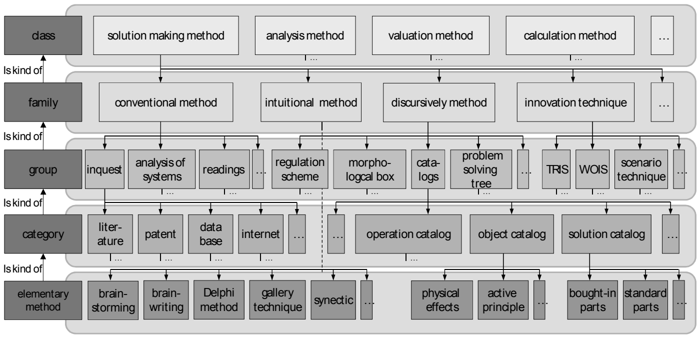


Figure 3. Structuring levels of methods

Some methods such as quality function development (QFD) are only performable by the combination of several elementary methods [Birkhofer07]. Basic methods, which stand on the highest specificity level and their combinations, are to be made available to engineers in the process navigator in correspondence with the selected DfX strategy.

2.3 Model integration

In order to get a suitable method support for DfX and to be able to integrate it into the process navigator, the DfX model and the methods structure must be unified and methods must be related to the DfX criteria. It has been found that the DfX criteria can be integrated into phases of the product development process by matching activities, results and contents of the product life phases to their corresponding DfX criteria. Fig. 4 shows an example for the design phase. In early phases of the process, general influences and DfX aspects are taken into consideration, whereas more and more concrete levels are used with the progression in process execution. Basic instructions are therefore considered only in the latest development phases.

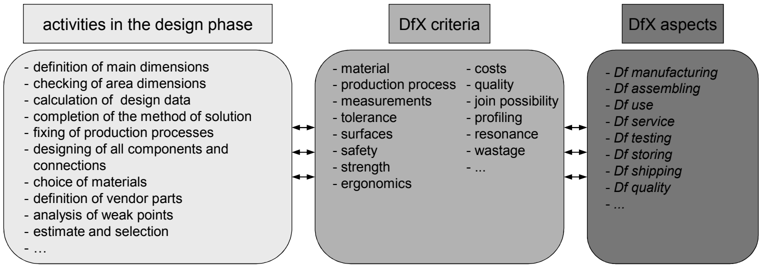


Figure 4. Classification of DfX criteria in design phase

It turned out that a method can be used for the implementation of several DfX criteria. Thereby not the way the method is executed, but the viewpoint on the method is changed. For evaluation methods, for example, certain criteria have to be specified, which highly depend on the choice of the respective DfX strategy. If low costs are fixed for a product, costs will have a very high value while assessing a certain solution. However if the environmental compatibility is highly valuable for the company philosophy, this method (evaluation) will be treated differently, i.e. the solution will be evaluated under the environment viewpoint.
A coupling of the DfX structure and the method structure is only meaningful on the concrete level of the DfX criteria, since the upper levels are too general. The choice of methods depends on several factors. First, this highly depends on the considered partner system "X" and on the DfX criterion, because appropriate focuses (e.g. costs, manufacturing or environment), implemented in different steps of the PDP, depend on the chosen DfX strategy. In addition method selection is affected by the existing information and data created during the process. Moreover the selection of a method also depends on the current design situation and activity of engineers. Depending on the chosen DfX criteria and DfX strategy, different influences affect the process planning and execution. Therefore the process is directly and highly affected by the implementation of DfX aspects. For instance, when the evaluation of solution alternatives must be accomplished, various evaluation methods are available in the method structure depending on the objective function. These methods are used depending on existing data in the PDP.

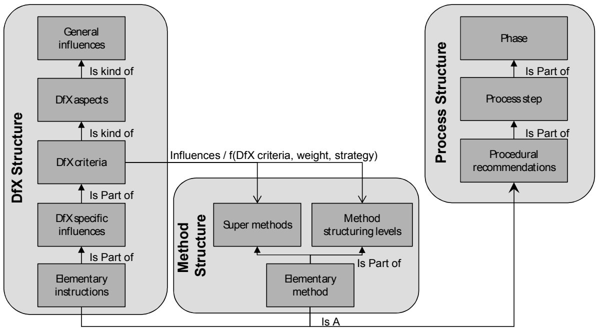


Figure 5. Coupling of hierarchical DfX structuring with method structuring

This example demonstrates well that a method (in this case, the evaluation of a solution) highly depends on the partner system "X". In case "costs" was selected as DfX criterion for the strategy "designing for customer", the evaluation criteria of the appropriate method are strongly affected by "Design for Cost". The classification of very concrete DfX instructions takes place on the *elementary instructions'* level, which contain specific DfX knowledge and have procedural instructions in the appropriate design situations for engineers depending on the PDP phase. As per fig. 2, these *elementary instructions* constitute the lowest level of the DfX model. Fig. 5 highlights this connection.

In order to connect the DfX model with the process model, in addition to phases and process steps, which are both part of the functional aspect, procedural recommendations are introduced. However these are not directly represented in the process model but provide a placeholder for elementary instructions and elementary methods. In line with the generalization/specialization of classes in object-oriented programming both elementary instructions and elementary methods can be used and replace procedural recommendations.

3. Technical Implementation

After presenting the concepts to define DfX- and process models and how they can be integrated, we will now describe their implementation in the Knowledge Base and the Process Navigator. For this purpose we will provide a detailed data model and outline the user interface.

3.1 Integration in the Knowledge Base

The Knowledge Base, which was developed in this project, expands the concept of a multidimensional process-oriented knowledge management system presented in [Jablonski02]. Its knowledge space is divided into separated orthogonal dimensions related together by defining the set of possible semantic relationships which can hold between the various concepts of each dimensions. This kind of specification constitutes an ontology [McGuinness02]. Its implementation is based on Semantic Web technologies, in particular the Ontology Web Language (OWL) [Antoniou03]. The formalism used for the data model relies on Description Logics [Baader03] family which allows a clear representation of the knowledge and provides reasoning possibilities over the data. Thus, not only the hierarchical relationships and classification of the various concepts can be modelled, but it is also possible to model and evaluate any kind of relation between them. This flexibility considerably facilitates the integration of the different data models.
The Knowledge Base has to store all the information that is presented to the engineer in the Process Navigator. It is able to store information and data to support the whole PDP, especially data structures for the perspectives defined in the process model are provided (i.e. data or operational perspective).
Figure 6 shows an extract of the ontology for the FORFLOW knowledge base. The relations between the DfX-Structure (on the left side of the figure), the methods structure (on the centre, simplified in this schema) and the process model (on the right side) can be identified. The method tree is represented using sub/super relationship between methods from each level. The ontology usage is depicted with the example instances we added for the most important concepts.

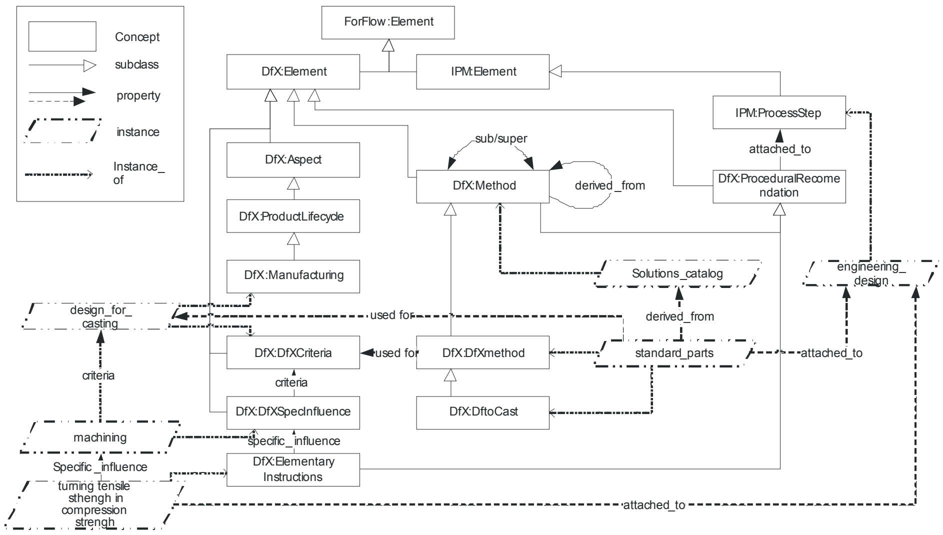


Figure 6. Ontology model for DfX criteria

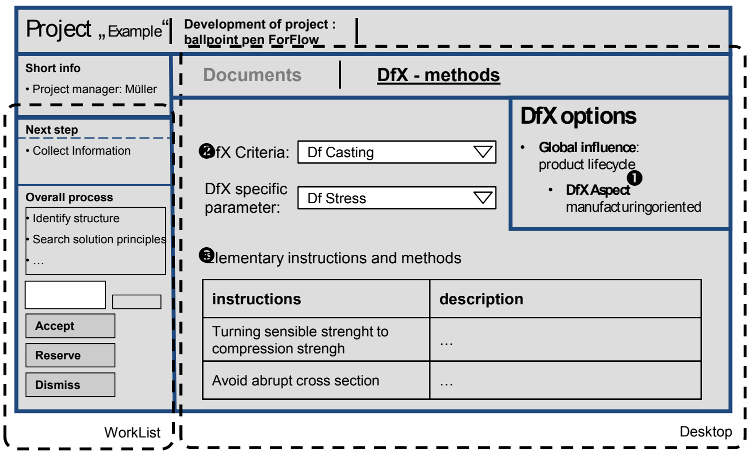
The definition of a method under a certain DfX-Criteria (e.g., the use of a method in relation to Df casting) can be expressed using the description logic formalism in the following definition:

$$method\_for\_casting \equiv \text{method} \cap \exists used\_for.DfX\_casting \tag{1}$$

This relationship can be used and understood by the software reasoner to automatically classify the various methods into their respective categories and infer relations between them. Other statements more or less similar to this example can be defined in line with the presented one.

3.2 Integration of DfX-Criteria into the process navigator

The Process Navigator is the user interface to process management system and builds on top of classical workflow principles [Jablonski96]. However these concepts have been modified so that navigation through the development process is maintained while the developer gained the flexibility to decide about the next stops. This flexibility is a requirement for the process navigator in order not to restrain the creativity of developers.
After selecting a step from the worklist, information about this step is displayed in the details view (DESKTOP). In this respect, the Process Navigator differs from conventional workflow systems as no application (e.g. a CAD program or text editor) is called directly, but a context sensitive information system. The context is determined automatically by the selected work step. The DESKTOP is partitioned into several topic sections where the information on product development is being displayed. Besides a section for DfX requirements (presented below) others covering different dimensions of the Knowledge Base (e.g. data perspective or operational perspective) are also available. At the beginning of each PDP, the project manager and stakeholders define the DfX strategy (e.g. design for cost or lightweight design) and thus the relevant DfX criteria. Starting with the DfX strategy, from the multitude of available DfX criteria, those which need to be fulfilled in the PDP are selected and weighted. Selection and weighting of DfX criteria influences the information displayed in the process navigator.
In Figure 7 a simplified layout of the process navigator is shown. The DfX selection field ❶ presents previously selected general influences and DfX Aspects. They are directly loaded from a knowledge base where the project status is stored and can only be altered by authorized users. A user can select the DfX criteria (linked to the DfX aspects) in the two drop down boxes ❷. Then in ❸ the relevant elementary instructions and methods are displayed, which support developers directly in their work. Only the elementary instructions and methods directly applicable for the selected DfX influences and aspects are shown. This way it is ensured that the user get only correct information.


Project „Example“
Development of project : ballpoint pen ForFlow
Short info
• Project manager: Müller
Next step
• Collect Information
Overall process
• Identify structure
• Search solution principles
• …
Accept
Reserve
Dismiss
WorkList
Documents
DfX - methods
DfX options
• Global influence: product lifecycle
❶
• DfX Aspect manufacturingoriented
Desktop

❷DfX Criteria: Df Casting

DfX specific parameter: Df Stress

❸Elementary instructions and methods

instructions	description
Turning sensible strenght to compression strengh	…
Avoid abrupt cross section	…

Figure 7. FORFLOW Cockpit

4. Conclusions

The aim of this contribution was to develop partial models for the DfX structuring, DfX methods and a process model and integrate them into a single comprehensive one. With the definition of an integrated data model and the derived ontology for the process based storage of DfX criteria, this goal was reached. By storing DfX criteria in the ontology, the relevance of the concepts can be assessed by a computer and they can be displayed in the process navigator at the right time. This feature contributes significantly to supporting developers in their work, as DfX criteria and methods now can be automatically analyzed and classified by computer systems.

Acknowledgements

The research work described in this paper conducted in the Bavarian research cooperation for "Process- and Workflow Support for Planning and Controlling Procedures in the Product Development" supported by the Bayerische Forschungsstiftung by the Chair of Applied Computer Science IV (University in Bayreuth) and the Chair of Engineering Design (University of Erlangen).

References

Abecker, A.; Hinkelmann, K.; Maus, H.; Müller, H.J.: Geschäftsprozessorientiertes Wissensmanagement. Springer-Verlag, Berlin 2002

Antoniou, G., van Harmelen, F.; Web Ontology Language: OWL. In: Staab S., Studer R., eds, Handbook on Ontologies in Information Systems, Springer-Verlag, 2003.

Baader F, Horrocks I, Sattler U. Description logics as ontology languages for the Semantic Web. In: Hutter D, Stephan W, editors. Festschrift in honor of Jörg Siekmann: Springer; 2003.

Bauer, S.: Design for X–Ansätze zur Definition und Strukturierung, In: Design for X, Beiträge zum 14. Symposium, Neukirchen, 13. und 14. Oktober 2003, Hrsg.: Meerkamm, H., Lehrstuhl für Konstruktionstechnik, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen: 2003, S. 1-8

Birkhofer, H.: elementary design methods and their benefits for research and practice, in Bocquet, Mekhilef, Duffy (Hrsg.): ICED 2007, Paris 2007, paper ID 15

Jablonski, S, Bussler, C.: Workflow Management – Modeling Concepts, Architecture and Implementation. London: International Thomson Computer Press, 1996.

Jablonski, S., Horn, S., Schlundt, M.: Prozessorientiertes Wissensmanagement mit der i>WorkBench. In:

McGuinness, D.: Ontologies come of age. In Fensel, D. et al. (eds): The Semantic Web: Why, What, and How. MIT Press 2002.

N.N.: VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. In: VDI-Handbuch Konstruktion, Berlin. Beuth Verlag GmbH Berlin Düsseldorf, 1993.

Roth, K.: Konstruieren mit Konstruktionskatalogen – Band I - Konstruktionslehre. 2. Auflage, Springer Verlag Berlin Heidelberg,1994.

Rude, S.: Wissensbasiertes Konstruieren, Zugl.: Karlsruhe, Univ., Habil.-Schr., 1998, Aachen: Shaker Verlag, 1998, Kapitel 2

Wallmeier, S.: Potenziale in der Produktentwicklung; Möglichkeiten und Grenzen von Tätigkeitsanalyse und Reflexion; (Fortschr.-Ber., VDI Reihe 1: Konstruktionstechnik / Maschinenelemente, Nr. 352), Diss., 2001, Düsseldorf: VDI-Verlag, 2001

Dipl.-Inf. Matthias Faerber
University of Bayreuth
Department for Computer Science IV
Universitätsstr 30, 95440 Bayreuth, Germany
Tel.: +49-(0)921-557342
Fax.: +49-(0)921-557339
Email: matthias.faerber@uni-bayreuth.de
URL: http://www.ai4. uni-bayreuth.de