INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

THE PROCESSNAVIGATOR – FLEXIBLE PROCESS EXECUTION FOR PRODUCT DEVELOPMENT PROJECTS

Matthias Faerber, Stephanie Meerkamm and Stefan Jablonski
University of Bayreuth, Germany

ABSTRACT
The development of new, innovative products is a companies' fundament to achieve success in the long term. Processes have been successful to structure development processes, however no practical approach for a process management system that supports users rather than restricting them is available. Based on the process all influence factors such as human beings, tools, methods or quality aspects have to be integrated in a goal-oriented way. In this paper we show how a product development process can be supported by a process execution environment. First we introduce the requirements regarding the users and the quality perspective. Mapping these in a process model they are then implemented in a novel integrated execution environment, named ProcessNavigator. We focus the connection between the process model and the execution environment and show how engineers can benefit from a flexible process management system. Scenarios are used throughout the paper to demonstrate how users can interact with the system.

Keywords: process management, integrated product development, context sensitive process support

1 INTRODUCTION AND MOTIVATION

1.1 Introduction
The development of new, innovative products is generally one of the key factors for the success of any company. Only companies that are able to constantly develop new and qualitative good products are able to succeed on the long term. However modern product development is a challenging task [9]. Lots of influencing factors, such as costumer requirements, available resources, regulations and laws, have to be taken into account and mapped with the appropriate tools or design methods. The local distribution of today's companies is another substantial factor. It is indispensable that the teams located in different sites cooperate closely. For this reason the development tasks have to be coordinated between these teams and their results have to be combined.
It is well known that most companies use processes to structure and document their product development. A lot of publications exits on this topic (cf. [1, 3, 16]). The motivation to document its processes may differ from company to company. But a well modeled process helps companies to understand their business. It can be used to guide engineers through the development process or it can be used to improve the process itself. Yet, in many cases the process is still only used in static applications (e.g. handbooks or a process portal on the intranet [12, 15]). Although these static applications contain all required information, the users still have to search for information by themselves. A handbook or intranet portal is not able to actively propose tasks to the user depending on the status of the project or product development situation.
Over the past years, several approaches to assist employees with dynamic information systems have been studied and developed. Workflow management systems (WfMS) [6, 7, 14] for example can be used to support employees in executing clearly structured and repetitive tasks (i.e. the accounting of travel expenses). Usually such processes need standardized human interactions, like approving a sum, and therefore can be automated well. Moreover the order of the work steps is the same at all times (sum of expenses has to be checked before it can be approved) and it never deviates from the predefined process.

Conventional WfMS were designed to support exactly this kind of use case. Their aim is to automate well defined processes, automatically launch the correct application and provide all necessary input data to the user. Conventional WfMS allow no deviation from the predefined process and treat them as exceptions which must be solved without systems support. This way it is ensured that all processes are handled exactly the same way and no tasks are being omitted. Creativity and flexibility is not required (or desired) and therefore forbidden by these systems.

Although conventional WfMS systems have proven to be useful in the area of administrative processes, they cannot be used to support product development processes. The main reasons were already identified in [6] where product development processes are called “ad hoc workflows”: they typically involve human interaction and the sequence of task is often determined while executing them.. Designed to enforce strict processes, conventional WfMS would constrain engineers too much and thus suppress the creativity that is needed in modern product development processes.

In order to support engineers in developing new products the FORFLOW research association looked into existing workflow concepts to adapt them for product development processes. A new process management system (ProcessNavigator) was developed as a central information system which has to support and coordinate activities in the product development processes.

1.2 Motivation – typical product development scenario

We will further motivate the need for a process management system by introducing two scenarios that are typical for product development processes. These scenarios regard situations in the constructions phase and describe common situations that engineers can encounter. They will be used throughout the paper to highlight aspects of the ProcessNavigator or in order to explain its concepts.

- *Scenario 1*: After the *identification of the problem* and *deployment of the function structure* the product developer has to find the right *active principle* for a given set of DfX criteria. Because quite a number of active principles exist for a certain aspects, it is difficult to find the appropriate one for the individual development circumstances. Lots of influencing factors have to be mapped with the function structure and the active principles. In addition, interdependencies between DfX aspects have to be taken into account.

- *Scenario 2:* After the *concretion of principal solution alternatives* the engineer has to conduct an *evaluation according to economical and technical criteria.* As the technical solution seems to be the bigger challenge to the engineer, he decides to first start with an evaluation of the technical criteria and put off the economical evaluation although in the company’s general product development process, the economical evaluation is scheduled first. Nevertheless, to assure the quality of the product with respect to the economical aspect as well, this work step has to be executed later on.

Although these two scenarios are used throughout the paper to highlight some aspects, the ProcessNavigator was developed independently and can be used to guide engineers through a wide range of situations.

In this paper we will introduce the basic concept for this process management system. We show how flexibility for the process execution is provided by the ProcessNavigator and how the system supports users in different product development situations. In Section 2 will provide a list of requirements regarding the ProcessNavigator. They have been identified while developing the system together with our partners (both from industry and universities) in the FORFLOW project. Section 3 introduces our approach whereas Section 4 and 5 provide information in-depth. Section 7 summarizes and concludes the paper.

2 REQUIREMENTS

The requirements of a process management system to support product development process differ considerably from those of conventional WfMS. Although there are some common aspects, the main focus of such a process management system lies on flexibility and adaptability. This section gives an overview of the core requirements we identified in the FORFLOW project. We group the requirements into two sections. The first one describes the user related requirements; they mainly concern the user’s interaction with the system and the information he needs in a project. The second group is not directly linked with the user. It rather concerns requirements that can be related to project or quality management.

2.1 User perspective

The aim of the ProcessNavigator is to provide systematic support for engineers in product development projects. Unlike conventional WfMS the direct integration of applications is not the central concern of the ProcessNavigator. Instead, it aims at providing comprehensive information to the engineer, who has to know how a certain work step in development project has to be executed and needs guidance through the product development process. For this purpose, the ProcessNavigator has to provide the following information:

1. *Information about a work step*: The ProcessNavigator has to inform about the current work step. In particularly, this includes a detailed description of the work to be executed (i.e. operating instructions). But also other information (e.g. methods or DfX aspects [4]) that should be considered by engineers has to be provided.
   Here we can refer to Scenario 1 (Identification of active principles): Due to the amount of data and the limited capacity or time of engineers it is usually cumbersome to search for possible active principles and to keep in mind all possible interdependencies between them. In such a case a system is needed that provides engineers with a catalogue of available and applicable active principles. For this purpose, active principles have to be analyzed by a domain expert with respect to the implementation of different possible functions and transformed into an electronic catalogue with defined search criteria. Provided this data analysis was performed well, the system can offer appropriate active principles to developers. Nevertheless the engineer will still have to check them thoroughly, if they really suit to the problem.

2. *Documents and product models*: For each process step the ProcessNavigator has to provide input and output documents. Engineers should have direct access via the ProcessNavigator user interface. This saves engineers from searching for documents (e.g. in file shares) and provides them with the recent documents. It also has to provide templates and documents, which are created in a work step, which should be automatically archived by the system.
   In the context of Scenario 1 this means that starting the conception phase with the identification of the problem, the engineer needs information about the results finally desired. That is he needs a list of requirements describing the problem and possible approaches.

3. *Tools and systems*: The ProcessNavigator should inform about tools and systems that can be used in a work step. In particular this includes tutorials and how-to manuals of tools or software systems which should be used in the process step. Furthermore he needs information about the interdependence concerning other tools he can use in the process step.

4. *Decision support*: The ProcessNavigator has to support the user when alternative approaches to create a product are possible. The system should guide the user through the decision process giving arguments and information. After the decision was taken by engineers the system has to document it, which includes advantages and disadvantages as well. The documentation can be used afterwards to reconstruct how this decision was reached, fostering a more transparent product development process.
   This requirement can be connected very well to Scenario 1 (Identification of active principles) again: While looking for active principles in most cases several solutions are possible. Each of them has certain advantages and disadvantages, but finally the engineer has to decide for a single one. In this situation first of all, it would be useful for the engineer to get information about the consequence of the active principle regarding the work steps still following; furthermore he needs information about the interdependence concerning the other selected active principles.

5. *Support for milestones*: Product development processes are usually split into several phases which are ended by milestones. These are situations in the product development process where documents have reached a certain maturity. They are stable against changes so that the next

phases can build on these documents. After having completed a milestone in a process, the contents of the documents must not be altered.

An aspect which is especially important for the support of product development processes is the execution semantics of the process management system. It defines the process steps which can be executed by a user and in which order this can be done. It also defines how far users can deviate from the predefined execution sequence. The execution semantics regarding product development processes differs fundamentally from that of conventional workflow systems For the latter, the execution semantics is clearly defined by the given order of work steps in the process (cf. [6, 14]). A short example illustrates this:

> The following process is defined: A→ B → C. In a conventional WfMS, step B can only be executed if A has been finished. Similarly, B has to be finished before C can be started. Deviating from this predefined order or executing a step more than once is not possible.

However this execution semantics cannot be used in product development processes. It is too strict and limits engineers at their work. For this reason the requirements described above must be amended with requirements concerning the execution semantics as followed:

6. *Suggesting next steps*: The ProcessNavigator has to suggest the next steps that have to be executed to the engineer. This requirement is almost identical to the behavior of conventional WfMS. Engineers must be informed, which steps can be executed next and which steps are scheduled for execution in the following.

7. *Repeating steps*: Product development is not a linear sequence of steps. It is rather characterized by iterations and loops. Above all, small iterations occur without planning. Therefore the ProcessNavigator has to allow the engineer to perform these iterations, which includes the reenactment of already completed work steps.

8. *Skipping steps*: Creativity of engineers is a crucial factor of success concerning the product development. It must not be suppressed by a WfMS. Besides a product development process cannot always be planned completely in advance. Therefore engineers must be able to skip steps for the time being and execute them later. However skipped steps must not be marked as executed as they have to be offered again to the engineer.
   Referring to Scenario 2 (Evaluation of technical aspects previous to the economical aspects) this means: The ProcessNavigator should enable engineers to skip steps. For this purpose engineers needs to see both, the very next work step and these, which have to be executed later on. Depending on the current design situation he can then decide whether he wants to follow the recommendations given in the ProcessNavigator or skip the scheduled step (*economic evaluation*) to first execute the step *technical evaluation*. However, the ProcessNavigator has to keep a record of all skipped steps and propose them to the engineers for later execution.

9. *Omitting steps*: In rare cases it appears that work steps do not have to be executed, although they have been scheduled in the product development plan. The decision, whether a process steps has to be executed or not, is complex and can only be taken by engineers. The ProcessNavigator has to support these situations and display the consequences of omitting a work step (i.e. it has to show which other work steps depend on the output data). When the decision omitting a process steps was taken, engineers have to document their decision so that it can be retraced later on.

All in all, the ProcessNavigator has to be a system that supports engineers during the product development and provides hints, which steps and tasks have to be executed next. Information has to be provided regarding the way a step has to be executed and what has to be taken into account. However the system must not constrain engineers giving them guidelines which are too strict. It has to provide enough flexibility needed to finish the project successfully.

2.2 Process or quality management perspective

The previous section discussed requirements that directly affect the engineers developing a certain product and how the ProcessNavigator can support them. However, user interaction with the process management system is only one part of the system's requirements. In modern companies not only engineers affect product development processes, but also the companies' business strategy and in some cases legal requirements influence it. A process management system must be able to cope with them; furthermore quality standards like the ISO 9001 or CMMI were introduced in many companies. One of the aims is to standardize and institutionalize process in companies. The ProcessNavigator has to support these goals, too. Therefore we add the following requirements to the ones listed before:

10. *Integration into existing process landscape*: Most companies have already defined product development processes. They were defined for the purpose of process documentation or finding approaches for improvements. Also quality management standards (e.g. ISO 9001) require the definition of a process landscape. The ProcessNavigator must be able interpret and execute them to finally leverage these processes.

11. *Enabling process evolution*: Due to lessons learned or due to improvement programs, processes have to be updated and adapted to new situations. The ProcessNavigator must be able to implement these changes easily.

12. *Documentation of process execution*: The ProcessNavigator must document process execution. On the one hand process documentation is required by quality management standards; on the other hand it also offers the possibility to trace a product development process and find shortcomings.

The requirements listed in this section can only give a short overview. However some central aspects could already be demonstrated. The ProcessNavigator has to support engineers and users by providing them with all necessary information about a product development process. It has to offer maximum flexibility so that only a minimum of needed constraints exists. Only this way the creativity, which is important for an optimal result of product development processes, can be maintained. In addition to support engineers, the ProcessNavigator can also help process management departments establishing process in a company and institutionalize them in a stable state. In the next sections we show how these requirements can be implemented by a process management system.

3 GENERAL APPROACH

To implement the requirements listed above, we have designed the system as a process based information system which provides continuous and integrated support for engineers during product development. In this context continuous and integrated means, that all aspects that are relevant for a product development process must be derived from a central model, which is the basis for the execution in the ProcessNavigator. Aspects of the product development process can be easily traced starting with the process model up to the information system.

Figure 1 depicts this approach. On the left side, the various aspects of the product development process can be seen. In the FORFLOW project we concentrated on the process containing work steps, data and product models, systems and tools as well as the people which are involved in the process. All these aspects have to be integrated into one model. However our basic approach is not limited to the aspects just mentioned. By extending the underlying model and integrating new features, the ProcessNavigator can be easily adapted to new requirements.

The process model provides a framework which can be interpreted by the ProcessNavigator accordingly. Therefore all the elements of the process model can finally be displayed in the ProcessNavigator and engineers can then be informed about possible next steps, input or output data as well as methods. All this is necessary with respect to the required flexible execution semantics.

Mapping all aspects to one common model (the process model) has yet another advantage. As the ProcessNavigator derives all information from the process model, the system can be easily adapted to new development processes. Instead of changing the ProcessNavigator's software, the underlying process model has to be modified which has an impact on the work steps displayed in the ProcessNavigator.

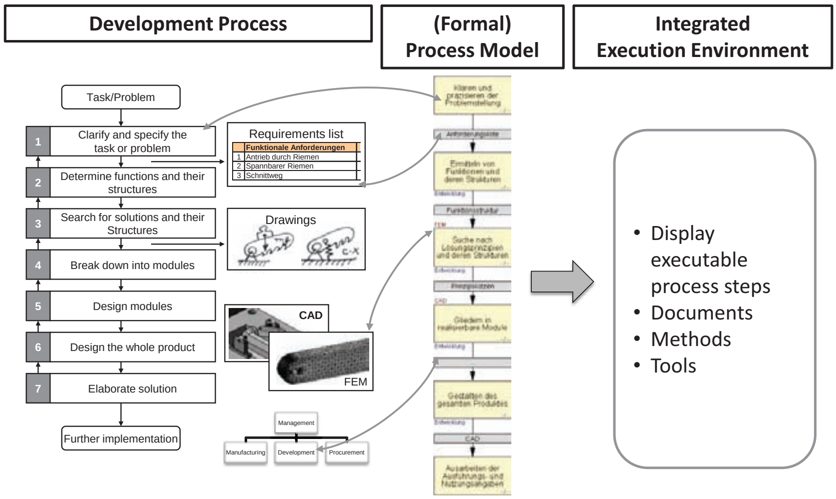

Figure 1. General solution approach

One of the requirements (req. #11 - Integration into existing process landscape) demands that the ProcessNavigator fits into an existing process landscape. By using a process model as basis for the ProcessNavigator this requirement is met. This way, the ProcessNavigator can leverage on existing work and it supports existing processes instead of introducing new ones. Although we will introduce the perspective oriented modeling approach in the next section and use it for modeling our processes, any well defined process model can be used as basis for the ProcessNavigator. Also requirement #12 (process evolution) can be easily implemented by changing and adapting the process model.

4 PROCESS MODEL

The process model is the fundament for the process execution using the ProcessNavigator. For the modeling of the product development process we apply the perspective oriented modeling approach [7]. The main design goal of this method was to develop a flexible system that can be easily extended and adapted to new requirements. This can be achieved by modeling processes using different viewpoints – so-called perspectives. Each perspective highlights a different part of a process. Although all perspectives can be treated isolated and may in fact build up their own model, only all perspectives combined constitute a complete description of a business process. Generally the following five perspectives are basis for each model:

- *Functional perspective*: With the functional perspective all work steps which have to be executed during the whole process are defined. Each work step describes a well-defined task which is necessary to develop the product (requirements #1 and #4).

- *Behavior-oriented perspective*: This aspect defines the causal sequence of the work steps. Mainly predecessor and successor relationships are identified and defined in this perspective (requirements #4 and #6).

- *Data-oriented perspective*: Data and product models, which are needed regarding the product development process, are defined here (e.g. the product's requirements list, concept outlines, CAD-documents etc.). Furthermore input and output documents of each work step are specified. From this definition, a data flow can be derived (requirement #2), relating outputs to inputs.

- *Operational perspective*: This perspective describes tools and systems which are used in the process. Regarding product development, these are for example CAD programs, FEM programs etc. (requirement #3).

- *Organizational perspective*: With this perspective the responsible persons (engineers) for the execution of a work step are defined.

Figure 2 illustrates a cut-out of a product development process. It was taken from the description of a design process in [10] and depicts the conceptual design of a component, starting with *abstract the circumstances to identify the problem*, via *combine the active principles to get a structure* to the point of *concretize to principal solution alternatives*.

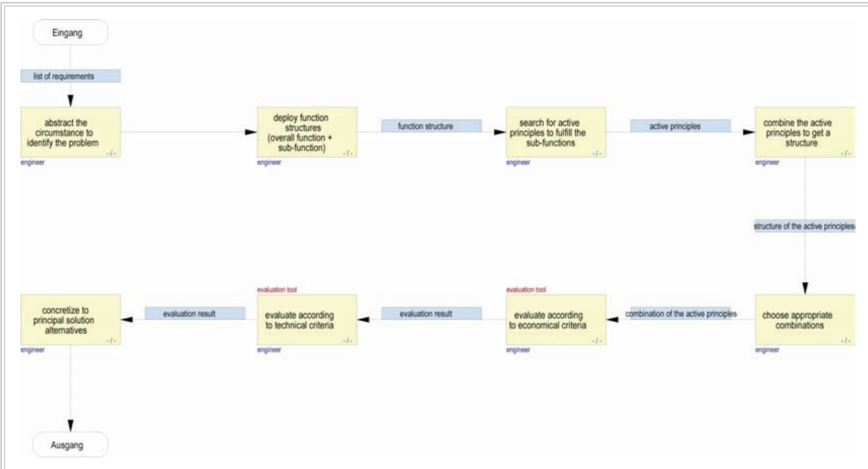

Figure 2. Process Model

In the process model the work steps (functional aspect) are connected by the so called control flow, which defines the order of the work steps. These are interpreted later by the ProcessNavigator. Furthermore input and output data of each work step are shown. The output data are created on the basis of the information offered by the input data and by means of the indicated tools.

In the context of the product development process some more perspectives have to be considered: ***Milestones*** and ***deciders***. Milestones are assigned to a special situation in the process where the documents have reached a defined level of maturity. Therefore the product developer can start with the processing of the following phase. Typical milestones are for example: *definition of the project concluded*, *concept concluded* etc. As in Figure 2 only a small cut-out is shown milestones and deciders are not included.

Deciders are used to model splits in the control flow. They are necessary in case the product developer has to decide how to carry on with the process. For example, after a review of a proposed solution, he has to decide whether the results of the conducted work steps comply with the requirements and the project team can keep on with the process; or, in the opposite case, whether the solution has to be revised.

Although each enterprise has to adapt the process model regarding to its respective circumstances, it is only an idealized mapping of the product development. In particular the process model does not comprise the required flexibility (requirements #7 and #8). The ProcessNavigator has to implement it during the execution of the process. However the necessary information concerning the work steps, the documents and the tools applied are available so that the ProcessNavigator only has to read them.

Like the scenarios presented in section 1.2, the process model in Figure 2 is only used for illustrating the concept of the ProcessNavigator. It is clear that referring to Scenario 2, the order of steps could also be changed (i.e. model the technical evaluation before the economical one). However, this discussion is not the aim of this paper. Instead we want to present a process management system (the ProcessNavigator), which is able to execute process flexibly and which is able to deviate from the control flow defined in the process model. In the next section the ProcessNavigator itself is described and we show how users can interact with the system.

5 PROCESS EXECUTION SYSTEM (PROCESSNAVIGATOR)

The ProcessNavigator is divided into a worklist, where the executable work steps are displayed, and an application (named FORFLOW Desktop), which offers information concerning the processing of a work step. Regarding this main structure, the ProcessNavigator is comparable with conventional Workflow Management Systems. But in several aspects worklist and desktop of the ProcessNavigator differ from ordinary WfMS due to the special requirements (see Section 2.1). Figure 3 shows the general layout of the ProcessNavigator.

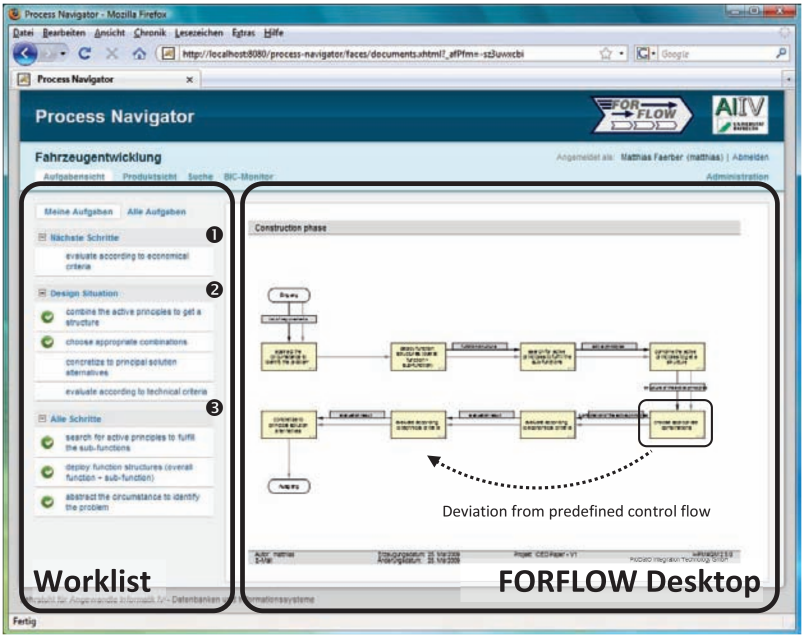

Figure 3. ProcessNavigator

The worklist displays work steps which can be executed by the user; they are derived from the functional perspective of the process model. However, other than in conventional WfMS not only the next, but all available work steps are displayed. Thus the flexibility can be provided, which is necessary for the execution of a product development process. In order to inform the users about the next executable work steps, the conventional work list is divided into several subsections. Each of them comes up with a special meaning (similar to a priority level) and may include more than one step. In the context of the FORFLOW project we decided to divide the worklist into three sections. It should be mentioned that for each company or use case the meaning and the content of the sections as well as the number of the sections can be configured individually. In our context the sections are specified as follows:

- ***Next steps (❶)***: The *Next steps* subsection shows all work steps, which are the next steps to be executed according to the control flow in the process model. Taking the example in Figure 3 the work steps up to the step "*choose appropriate combinations*" have been executed. Consequently the work step "*evaluate according to economical criteria*" is listed as the next executable work step in this section. In case the control flow splits due to a decider, more than one work step may be included here.

- ***Context steps (❷)*:** Here all the work steps are listed, which are assigned to the context of the actual work step. Context is perceived as a certain number of work steps, which can be executed before and after the current one. The engineer can use this list to go back to a work step that was executed recently (e.g. for a review) or to skip the next step temporarily (requirements #7 and

#8). This list is important as there a lots of small iterations and loops in development processes. Users can do such iterations by selecting a step from this list.

- ***Complete process (❸)***: This part of the work list contains all work steps of the over-all process. The user benefits from this list in case he needs to start a step that is not included in one of the other two worklist sections (requirements #7 and #8). But this should be an exception; normally the user chooses a work step from one of the sections *Next steps* or *Context steps* presented above.

Due to the fact that the developer can select any work step of the work list, the decision about the way he continues with the process is left to the developer. Therefore the most important requirement concerning the ProcessNavigator – flexible execution semantics – is implemented. However the section *Next steps* ensures that the user knows, which work step is planned to be executed next. By breaking down the work list into different sections and showing all work steps, the ProcessNavigator combines systematical navigation with flexible process execution.

In Scenario 2 a deviation from the predefined control flow is described. This deviation is shown in Figure 3 as a dotted arrow. Instead of following the control flow, the engineer wants to evaluate the technical criteria before economical criteria. After finishing the step "*choose appropriate combinations*" (arrow start), the worklist has the following setting (cf. Figure 2 for the model): *Next* – "*evaluate according to technical criteria*"; *Context* – "*combine the active principles …*", "*choose appropriate combinations*" and "*evaluate according to technical criteria*"; *Complete process* – all other steps. Instead of selecting the step from the *Next steps* list the engineer can choose the work step from the *Context steps* list. This way he can easily deviate from the underlying process model, as required by the design situation, without having to change the underlying process model.

After the selection of a work step in the worklist, the FORFLOW Desktop is activated. There users can find information concerning the current work step. All this information can be used to complete the work (requirement #1 and #3).
Figure 4 illustrates the ProcessNavigator after the selection of the work step "*choose appropriate combinations*". The FORFLOW Desktop itself is divided into several tabs. By selecting one of them on the upper side band (❹), the user activates the tab and sees the included content. In the current version of the ProcessNavigator the following tabs are available: *documents* (product model), *tools, methods, CAX data format* and two sections to search in an underlying knowledge management system (*knowledge base* and *CAD search*). Some of them are explained exemplarily.

- ***Documents***: This tab shows input and output data of the selected work step. They are derived from the data perspective of the process model and provide the user with all documents that are necessary to work on the task. It is also show, which data (documents) has to be created in this step. Also templates or examples of already completed documents can be included here.

- ***Tools***: This tab aims at supporting the user to select the right tool. This may be a difficult decision due to the complexity of the tools. Taking into account the context of the overall development process and the actual development situation the ProcessNavigator suggest suitable tools. It also shows dependencies between these tools regarding the data formats. For more information please refer to [17].

- ***Methods***: Here the user gets an overview about the DfX aspects he has to keep in mind during the execution of a work step. The placeholder "X" has to be filled with a special influence on the product. Meanwhile a lot of guidelines and methods for DfX exist. They help engineers to implement the particular features of the product and support the user in a target-oriented handling of the product development process. A detailed description of this tab and its integration into the ProcessNavigator is available in [5].

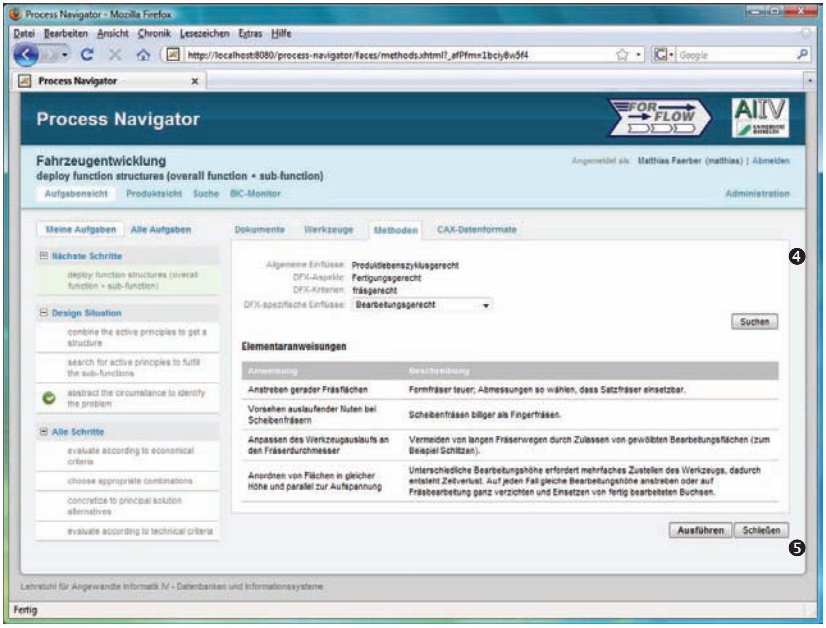

Figure 4. ProcessNavigator with DfX hints

After the user has created and uploaded all required documents, he can complete the work step (❺). Finishing a work step leads to an update of the work list. In particular, this affects the section *Next steps*. The currently finished step is removed from this list and the next step is displayed. Whereas in section *Context steps* only little updates, relative to the defined context are necessary. Beside to this the finished work step has to be tagged as executed (small marker in the work list). Finished work steps can be selected again by the user for example to review or revise the results of this step.

> Scenario 1 describes a situation where a user needs information about certain DfX criteria. As these depend on the current design situation (work step, project parameters, etc.) the ProcessNavigator is querying a data structure similar to a database system to retrieve the necessary information (the underlying mechanism is described in [5]). Within the tab *Methods* he can get all necessary information and can use this information. This works similar to a normal web search, but is actively provided by the ProcessNavigator depending on the design situation.

Milestones and *deciders* (requirements 4 and 5) can be displayed in the work list as well. To distinguish them from a normal work step they are flagged by the character "*M*" or "*D*" respectively. Selecting one of these items, the typical tabs of the FORFLOW Desktop are replaced with tabs adapted to the information necessary for checking a milestone or taking a decision. For the latter, the user gets background and context information regarding the actual situation. This helps to decide about the progress of the development process in case of alternatives. Furthermore the developer has to document his decision. This documentation can be used later on to trace decisions made in the process.

Selecting a milestone from the work list, the user gets an overview about all documents which have to be created up to this milestone. In case of having all documents he can complete the milestone. To assure data consistency and data integrity, it is not allowed to change the approved documents again; the ProcessNavigator prevents changes to these documents via the normal interface and users can only view them. In case of incompleteness or quality problems with the documents, the project manager has to revise these documents via a special interface.

In Section 2.2 process documentation is demanded as another requirement (#13). The ProcessNavigator implements this by maintaining a work step execution log. There all executed steps, their order and how often a certain step is selected is saved. This log can be used to prove that all required or planed steps have been executed (e.g. in QM appraisals) or it can be used for process improvement purposes. If for example a certain step is always omitted, this is an indicator that the underlying process is not ideal and may need improvement.

6 RELATED WORK

After having introduced our approach, we will have a short look at related work. We will present the concepts of an integrated process management system and give a short overview of product data management systems or product lifecycle systems (PDM/ PLM system).

In [2], the author presents an integrated process management system to support the coordination of tasks. Based on a modeled business process, a WfMS is used to schedule tasks and distribute work packages. For process modeling, extended UML activity diagrams are used. These activity diagrams are then transformed into a workflow execution language (*XPDL / BPEL*), which forms the basis for the process execution in the *PTC Windchill* workflow engine. This way, the system assigns task to users based on a predefined workflow. However, although the author presents no details on how users can interact with the system and thus how well the system provides support for product development processes, some conclusion can be drawn from the systems used: XPDL or BPEL are vendor neutral process definition and execution languages. Similar to programming languages, they define which steps have to be executed in a business process. These languages are also commonly called "orchestration languages," as their purpose is to organize the execution of a predefined plan. However, flexible execution,is not supported by these languages; moreover, the execution of the workflows is very strict. PTC (Parametric Technologie Corporation) Windchill [8], primarily, is a content and process management software to manage the data and the process during the overall product life cycle. The software is very comprehensive regarding its functionality and offers support during the whole value chain. Nevertheless, for process support a conventional workflow concept is used and work steps are provided in a conventional worklist.

Other approaches supporting users are product data management systems and product lifecycle management systems (PDM/ PLM systems). They are used to manage data, applications and processes [8, 11] during product development projects. The reduction of the development time, the increase of process and product quality and last but not least at the promotion of interdisciplinary team-work is one of their primary goals. All these are key factors regarding the general competitiveness of a company. For the connection of heterogeneous systems, they offer interoperability functions, e.g. possibilities for data exchange. Change management and configuration management, for example, are much easier to implement due to system integration and cross-company collaboration. Their advantages with respect to the appropriate context (data management) are unchallenged. Nevertheless, the specification of the process and the implementation of the PDM/ PLM systems are too strict in the context of product development processes. There, the developer should decide more flexibly and in terms of the design situation what he wants to do. The systems do not fit completely to the requirements regarding the product development, especially in terms of flexibility during the execution of tasks. However, the ProcessNavigator could doubtlessly leverage on their data management functions.

7 CONCLUSION

The aim of this paper was to present the concepts and the implementation of the FORFLOW ProcessNavigator. We demonstrated the rationale of this research by presenting our solution approach, a process management system able to execute processes flexibly. An example implementation, the ProcessNavigator, is verifying its feasibility. Since the end of 2007 multiple project partners in the FORFLOW project are testing the ProcessNavigator successfully. This field test [13]– although not perfect and complete since some implementations are still on their way – has convincingly demonstrated that the engineering domain could greatly be supported by a process management system that can perform processes according to multiple execution semantics, especially flexible execution semantics.

Coordination of the work steps and the support for the cooperation of engineers in development teams, which have always been the strengths of process management systems, can only become apparent if users accept the systems. The flexible execution support that is provided by the ProcessNavigator is a key factor for this acceptance. The process management system and the provided execution semantics now reflect the typical iterative character of developing projects and no longer restrict users.

Furthermore engineers quickly get an overview over the open tasks that have to be finished in order to reach the project goals. Furthermore it makes sure that at the end of the overall process all essential work steps are executed to guarantee the quality of the result and the product respectively.

REFERENCES

[1] Allweyer T., Geschäftsprozessmanagement – Strategie, Entwurf, Implementierung, Controlling. Herdecke: W3L-Verlag, 2005.

[2] Bahrami A., "Integrated process management: from planning to work execution," in Proceedings of the IEEE EEE05 International Workshop on Business Services Networks, Hong Kong, 2005.

[3] Becker J., Kugeler M., and Rosemann M., Prozessmanagement, 5 ed. Berlin: Springer, 2005.

[4] Ehrlenspiel K., Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit, 3 ed. München: Hanser Verlag 2007.

[5] Faerber M., Jochaud F., Stöber C., Jablonski S., and Meerkamm H., "Knowledge oriented Process Design for DfX," in 10th International Design Conference, Dubrovnik, 2008.

[6] Georgakopoulos D., Hornick M., and Sheth A., "An overview of workflow management: from process modeling to workflow automation infrastructure," Distributed and Parallel Databases, vol. 3 (2), pp. 119-153, 1995.

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

[8] Liu D.T. and Xu X.W., "A review of web-based product data management systems," Computers in Industry, vol. 44 (3), pp. 251-262, 2001.

[9] Meerkamm H., "Prozesse: eine aktuelle Herausforderungen in der Produktentwicklung," Konstruktion, vol. 9 p. 1, 2007.

[10] Pahl G., Beitz W., Feldhusen J., and Grote K.-H., Pahl/Beitz Konstruktionslehre - Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung: Springer, 2007.

[11] Rouibah K., Rouibah S., and van der Aalst W.M.P., "Combining workflow and PDM based on the workflow management coalition and STEP standards: the case of axalant," International Journal of Computer Integrated Manufacturing, vol. 20 (8), pp. 811 - 827, 2007.

[12] Schmelzer H.J. and Sesselmann W., Geschäftsprozessmanagement in der Praxis, 6 ed. München: Hanser, 2008.

[13] Sharafi A., "Pilotierung und Evaluierung eines Prozessnavigators," in 2. Ergebnisbericht FORFLOW 2008, H. Meerkamm and K. Paetzold, Eds. Erlangen, 2008.

[14] van der Aalst W.M.P., ter Hofstede A.H.M., Kiepuszewski B., and Barros A.P., "Workflow Patterns," Distributed and Parallel Databases, vol. 14 (1), pp. 5-51, 2003.

[15] Wagner K.W. and Käfer R., PQM - Prozessorientiertes Qualitätsmanagement, 4 ed. München: Hanser, 2008.

[16] Weske M., Process Management - Concepts, Languages, Architectures. Berlin: Springer, 2007.

[17] Zapf J., Troll A., Stöber C., Rieg F., and Meerkamm H., "Prozess- und Workflowuntersstützung - Methodische Kopplung von DfX-Strategien mit CAx-Prozessunterstützung zur Bereitstellung praxisorientierter Handlungsempfehlungen," in 19. Symposium Design for X, 2008.

Contact: Matthias Faerber
University of Bayreuth
Department of Applied Computer Science IV
Universitätsstr. 30, 95440 Bayreuth, Germany
E-mail: matthias.faerber@uni-bayreuth.de
Tel: +49 921 557626
URL: http://ai4.uni-bayreuth.de

Matthias is a researcher at the Chair of Applied Computer Systems IV at the University of Bayreuth. He researches in process management and quality management systems. He is a member of the FORFLOW research cooperation which aims at providing engineers with a flexible process management system.