INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15 -18, 2005

TOWARDS AN IT SUPPORTED FUZZY FRONT END OF PRODUCT DEVELOPMENT

Felix Nyffenegger, Nadja Jamali, Dr. Carmen Kobe, Prof. Dr. Markus Meier

Abstract

Today it is widely accepted that the early stages of product development play an important role in the success of product innovation. However, unlike other important business processes the early stages find only poor support by new information techniques. This might be due to the vague structured nature of the Front End process and the need for structure caused by IT. To overcome this conflict a framework was developed which aims to structure the Front End processes but on the same time keeps it flexible as the framework allows tailoring a generic process to the requirements of a particular product definition project. A software tool guides the user of this framework through the configuration of a project specific process model. The output of the software will in future be used to automatically customise a workflow engine and a collaborative platform.

Keywords: Fuzzy Front End, Innovation-Process, Collaboration, IT supported processes

1 Introduction

1.1 Situation

In the past years, a lot of attention has been spent to the early stages of the product development process. It is widely accepted that this early stages (in literature often referred as fuzzy Front End) play an important role in the success of product innovation [1-4]. But while efficiency and quality in other important business processes are improved by new supporting information and communication technologies, the fuzzy Front End is only poorly supported by these systems. An IT supported Front End process can facilitate local and international collaboration in product development tremendously, since it would allow the integration of collaboration participants already at the early stages of product concept and definition or even before, where new product ideas are born.

However, process supporting software systems are only as good as the underlying process. But the crucial task of defining and cultivating a Front End process is left to the company that is implementing such systems. SME's, which corner the western European markets to a high degree, often do not have a well defined Front End process and find difficulties in defining one due to lack of resources or absence of knowledge. Therefore, methods and tools are needed to assist companies in finding the optimal process for an innovation project and learn from projects done in the past to cultivate a best practice.

1.2 Purpose of the Paper

In this paper a Front End framework and its software implementation will be presented. The framework aims to structure Front End processes and helps to transfer generic knowledge about the pre-study-phase into a company specific process model and than allows efficiently transforming it into a pre-study-project.

This will be the necessary basis to support the pre-study-phase with new information technology and integrate it into software systems such as Idea Management or Product Lifecycle Management. The application of the presented framework shall also enable companies to comprise an organisational learning process about conducing pre-study projects.

Based on the work done it should be possible in future to automatically run a workflow engine and a collaborative platform for Front End projects.

1.3 Methods

In a descriptive study based on literature and interviews in the industry a state of the art situation of IT support of the early stages of product development was elaborated. This study focussed on one hand on the processes and tasks of the Front End on the other hand existing tools and systems to support business processes with IT were analysed.

Based on the results from this study the problem was refined and in a prescriptive phase we developed a framework of a dynamically structured Front End process. This framework was then implemented as a software prototype to prove the concept.

In future, a second descriptive study has to be carried out together with industry to prove the concept in a real world environment.

2 State of the Art

2.1 Models of Pre-Study Processes

There are different models describing the Front End e.g. Cooper or Rosenthal [1, 3]. These models assume the existence of an optimal Front End process. But the adoption of these generic models to companies or to one particular pre-study-project seems to be difficult, especially as there are many iterative and interdisciplinary tasks [4-6]. Nobelius and Trygg showed in [6] that there might not be just one Front End process, but different versions of the Front End process. Depending on the type of project there are different sequences and priorities of the tasks to be done in the pre-study-phase. In [5] multiple models of the innovation process, e.g. the models of Cooper, Davenport or Herstatt were analysed. Mitterdorfer-Schaad concludes that most of these models can be reduced to the three phases of concept statement or preliminary assessment, product definition, and project planning, but they do not go more into detail. The goal of our research is to get one level deeper into detail by adding actually tasks to these phases.

The ZPE/ETH Reference model [7] defines the Pre-Study process as a sub process of the Front End process, which can be divided into sub processes that includes the tasks from developing an idea into a product design concept. This process must be fully understood prior to the establishment of the product requirements. The product concept includes the design draft, which translates the concept into a specified product. It includes feature definition, design considerations with respect to quality, capabilities, and preliminary performance and feasibility assessments. Marketing is an essential part of the product development phase, thus the tasks under this sub process create the foundation for planning through the collection of relevant marketing tactics and strategies.

2.2 Process supporting IT Tools

Today a wide range of business processes finds an integration of software systems to support or even partially automate them. Information technologies can reduce barriers in gathering, saving, and proceeding information and this way increase capabilities in terms of time and space. In Table 1 the most important and common concepts of process supporting tools and techniques will be briefly described. A business application always is a combination of several tools or concepts of the above list. In [8] such combined systems are referred as an *integrated process supporting system (IPSS)*. Most Relevant IPSS to Front End processes are Idea Management Systems and Product Lifecycle Managements Systems.

Table 1. Overview of Process supporting Software Systems

Application Class	Application	Description
Managing and Storing Information	Database Management Systems	A computer program designed to manage a large set of structured data, and run operations on the data requested by numerous users.
	Document Management	Centralised management of documents as well as regulation of access rights and shared use of documents.
	Workflow Management	The automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal [9].
Searching and Reporting	Search Engines	Search engines are able to find information out of unstructured information objects (e.g. a text document).
	Data-Warehouse Systems	Aggregation of data stored in several databases according to certain rules.
	Agents	Autonomous, proactive and reactive, software, which e.g. can collect information about a certain topic.
Collaboration	Groupware Systems	Collaborative workspace for project teams. Mostly asynchronous collaboration.
	Conferencing Systems	Allow real time communication between participants (e.g. Video Conferencing).
	Content Management Systems	A system used to organise and facilitate collaborative digital content creation.
	Whiteboards	Virtual whiteboards allow one or more persons to write or draw images on a simulated canvas.
	Application Sharing	Simultaneous usage of an application by two locally separated participants.
Planning Tools	Project Management Systems	Software which assists in planning and controlling projects.
	Process Management	Storing and communication processes, no execution.

2.3 Process Structuring Methods

For structuring complex processes Steward [10] suggests the methodology of the Design Structure Matrix (DSM). Design Structure Matrix (DSM) is a compact matrix representation of a system/project. The matrix contains a list of all constituent subsystems/activities and the corresponding information exchange and dependency patterns. That is, what information pieces (parameters) are required to start a certain activity and where does the information generated by the activity feed into (i.e. which other tasks within the matrix utilize the output information). The DSM provides insights about how to manage a complex system/project and highlights issues of information needs and requirements, task sequencing, and iterations. Its aim is to structure mutually affecting tasks in a process, into an optimised and banded sequence, by differently weighting the relations between tasks. While the so-called "tearing" or weighting of the relations needs to be done manually, the sequence of the tasks can be resolved automatically by a partitioning-algorithm.

2.4 IT support of the Fuzzy Front End of Product Development

A Benchmarking study [11] which focused on integrated process supporting systems for Front End processes showed that there are different solutions available on the market. These solutions can be subdivided into two categories: Systems that only deal with idea-selection and retrieval and product-definition solutions which support the whole Front End process from idea acquisition to decision point for the start of actual product development.

However, such systems as any IPSS are bound to a structured workflow. Dealing with such workflows is problematic especially during the dynamic phases of early product development [12]. To overcome this conflict between dynamic and iterative processes and the need for structure caused by IT systems, a more dynamic approach to structure processes and workflows is needed.

3 The dynamic Pre-Study Process Framework

In chapter two the conflict between the demand of an IT System for a more clear structure and the vague nature of Front End processes was mentioned. To solve this conflict we elaborated the approach of a dynamic Front End Framework and a method which helps to individually structure the process from an unstructured generic model to a project specific procedure for each pre-study project.

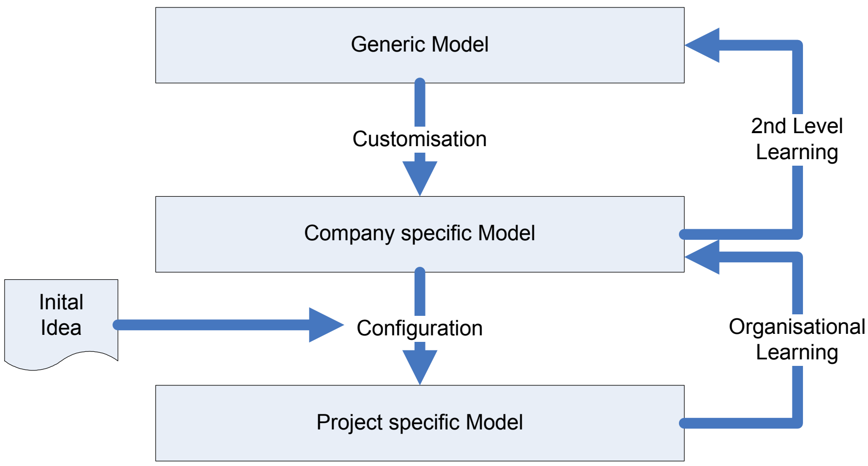

Figure 1. The Front End Framework

3.1 The Framework Overview

Figure 1 shows the Framework with its three models: the generic, the company specific and the project specific model. These levels can be considered as abstraction levels. In between processes are situated to bring a model from one level of abstraction to another. Customisation and configuration processes assist the transfer from the company strategic abstract models to a more concrete and specific one. Vice versa, the more generic models will be updated by learning processes

Each model is composed of the following elements:

Process	The process encapsulates a set of phases.
Phase	Phases bring a rough and sequential structure into the process. Two phases are always separated by a checkpoint which serves as a control and validation point of the state of a project.
Task	Tasks are sub-elements of a phase which describe one particular action to be done by one ore more persons. As the framework aims to sequence tasks situation-specifically, tasks in the model are not in any structure.
Information Object Type	Information Object Types define a structured unit (an entity) of information which has a name and certain attributes, as object types are defined in information sciences [13]. E.g. an information object type could have the name *Idea* and the attributes *Title*, *Description*, *Sketch* and *Risks*.
Input-Relation	Defines which type of information is used by a certain task.
Output-Relation	Defines which type of information is generated by a certain task.

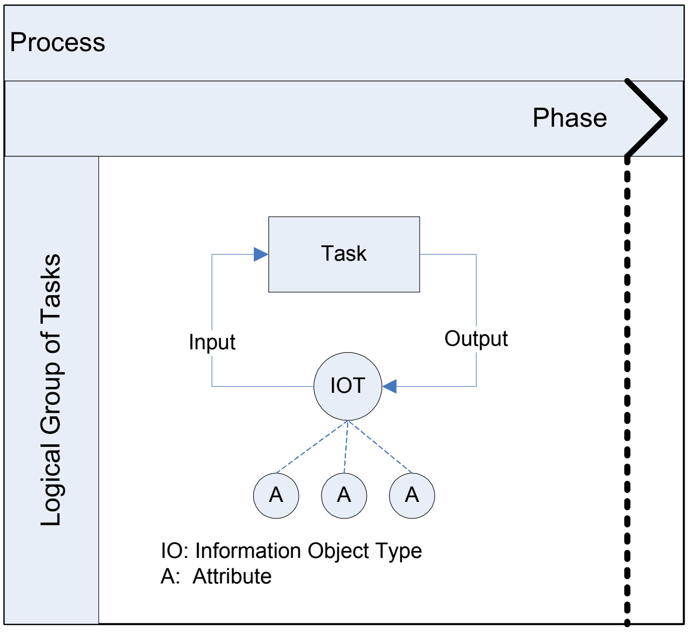

Figure 2. Elements of a Process Model

Both, input- and output-relations can serve as an indirect relation between two ore more tasks. One particular relation between two tasks can later, during the process of configuration, be weighted in order to define the degree of relevance of an input into a task. Figure 2 illustrates the elements in their coherence.

3.2 The generic Model

The generic model is the most abstract model in the Framework, which includes all processes, activities, and tasks that are relevant for the pre-study process. A checklist of about 65 processes, activities and tasks was formed based on a thorough literature search, the details are mentioned in [14]. A more concrete checklist is currently in consolidation with companies, as there are always tasks which are highly related to the functioning of a particular company. In contrast, this model is not even meant to be static, as the view to the pre-study process might change over time.

At this level of abstraction we do not define any input relations between tasks and information object types as in general nearly everything can be a relevant input to everything. What we define is only a draft of the information objects which result form a certain activity.

3.3 Customising the company specific Model

The company specific model represents the reduction of the generic model to a company model; this includes all the process, activities, and tasks essentially needed by a certain company and the development of a model, which is unique to the company and therefore not part of the generic model.

The information object types will be enhanced and completed to the company's needs. The structure of the information must be completely defined at this level as it will be the basis for project specific models and later the basis to compare different projects on the same structure of information. In addition, at this level we define dependencies between information and tasks. E.g. we define which information could be a possible input to another task. At this point the sequence of these tasks is not important, since the relevance for an input to a particular task depends mainly on the project itself.

Another element of this model is templates. Templates represent a set of preselected tasks according to one particular situation. E.g. the required tasks for a market driven product idea can be stored in a corresponding template. Templates play an important role in the organisational learning concept of the framework.

The process of transferring the model from the generic to the project specific level is called customisation. This process has to be done manually and can add up some effort. But it has only to be done once. And having the generic model as a basis and the structure of the model as a framework will help to be more efficient.

3.4 Configuration and the project specific model

The configuration process derives a project specific model from the company specific model. Main input to this process is the idea, defined by Kuehn as follows: *The most embriotic form of a new product or service. It consists of a high-level view of the solution envisioned for the problem identified by the opportunity.* [15]

Based on this idea either a project template or a new project specific model needs to be configured. Configuration means first the reduction of the number of tasks to the project needs and second bringing these tasks into a sequence. A sequence can consist of parallel, coupled, and sequent tasks. While reducing the tasks is a manual, intellectual process defining the optimal sequence can be done semi-automatically by using a DSM. For each cycle in the graph of task relations a priorisation of the inputs to a task has to be done. The sequence will be calculated automatically by partitioning and banding the matrix.

The output of the configuration process can be used to automatically generate a project plan and by adding resources it can be used to run a workflow system.

3.5 Organisational Learning at project reviews

One of the general benefits of applying this model will be to evolve best practice processes for different cases of pre-study projects. E.g. market driven or technology driven inputs to the pre-study process will result in different project specific models which both can represent a best practice process for its case. Thus company specific models remain dynamic. The end of every project if successful or not should be a review meeting. With the help of those reviews, the model will be updated with the experience gained during the project. This could mean to change the existing model or templates or to add new components e.g. templates to the model. Over time experience and tacit process knowledge will take influence on the model and a best practice can be approximated.

3.6 Second level learning for Software providers or Research Institutes

Experience of companies will – of course – also affect the generic model. In a so called second-level learning process such affection can be discovered and brought back to the generic model. This is not just highly interesting to research institutes but also to providers of idea and pre-study management software as it will help to continually improve their product.

4 Software Implementation of the Front End Framework

The framework described in the previous section can only be handled by adequate support of software. Especially the complex process of project configuration and the dynamic generation of a workflow need to be well supported. These are also the most promising components of the model in terms of time reduction. Thus an application was implemented which focuses on the configuration process.

The application is entirely written in C#, the background for this decision lies in the possibility of .NET to develop Windows and Web applications in the same environments and build on the same components. This will be crucial for future development.

4.1 The Architecture of the Software

The software is organised as shown in Figure 3. The base of the environment is the data representation of the company specific model and a data store which represents the project data as well as other relevant enterprise data such as information about employees. The company specific model is described in an XML dialect. To obtain maximum flexibility decision was made not to build on a standard process definition language such as BPML [16] or XPDL [17] but to develop a dialect on our own to exactly describe our model. An example of such a process definition file is given in Figure 4.

The central module and heart of the system is the *Front End Information Framework* which encapsulates all the meta-structure needed to describe and handle processes and information of the Fuzzy Front End. The company specific process model is read by the framework at startup time of the application. Besides the process structure elements the framework comprises two more modules which allow the automated export of information and projectplans to MSOffice™ applications. All other modules of the environment are based on this framework.

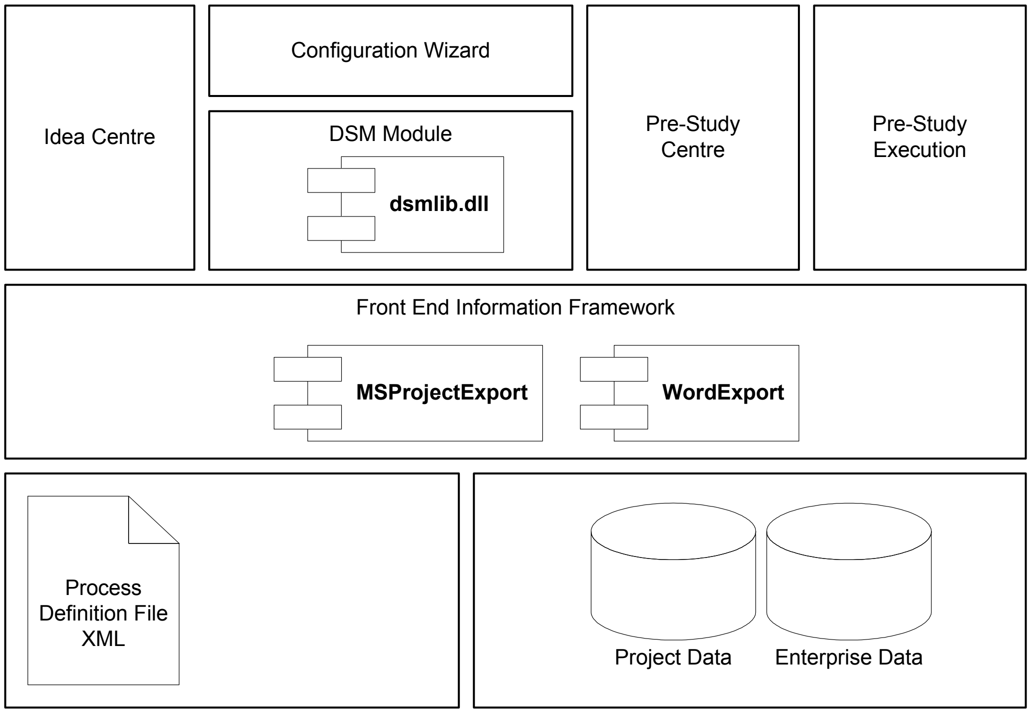

Figure 3. Software Architecture

Another very important module is the *DSM* Module, which handles the tearing and banding of the tasks. The module basically is an implementation of the algorithm presented by Kusiak in [18], to find the cycles in the tasks graph the deep search first algorithm is used. Around these basic algorithms the module implements an interface to the framework, so that we can feed the DSM with task objects from the framework.

On the top of the environment there are the GUI applications *Innovation Centre*, *Configuration Wizard*, *Pre-Study Centre* and *Pre-Study Execution*, which will be discussed in the following part of the chapter.

```
<?xml version="1.0" encoding="utf-8" ?>
<FIPML>
    <fipTypes>
        <fipType name="Idee" id="00.00001">
            <fipAttribute name="Titel" type="String" />
            <fipAttribute name="Slogan" type="String" />
        </fipType>
    </fipTypes>
    <fipProcesses>
        <fipProcess name="Simple Study" id="01.00001">
            <fipPhase name="Concept Plan">
                <fipTask name="Create Vision Statement" id="01.00002">
                    <fipInput type="Idee" />
                    <fipOutput type="VisionStatement" />
                </fipTask>
            </fipPhase>
        </fipProcess>
    </fipProcesses>
</FIPML>
```

Figure 4. Example of an xml Company specific Process Model

4.2 Functional Description of the Software

To better understand the configuration procedure and functionality of the software, in this section an exemplarily walk through a configuration process will be given.

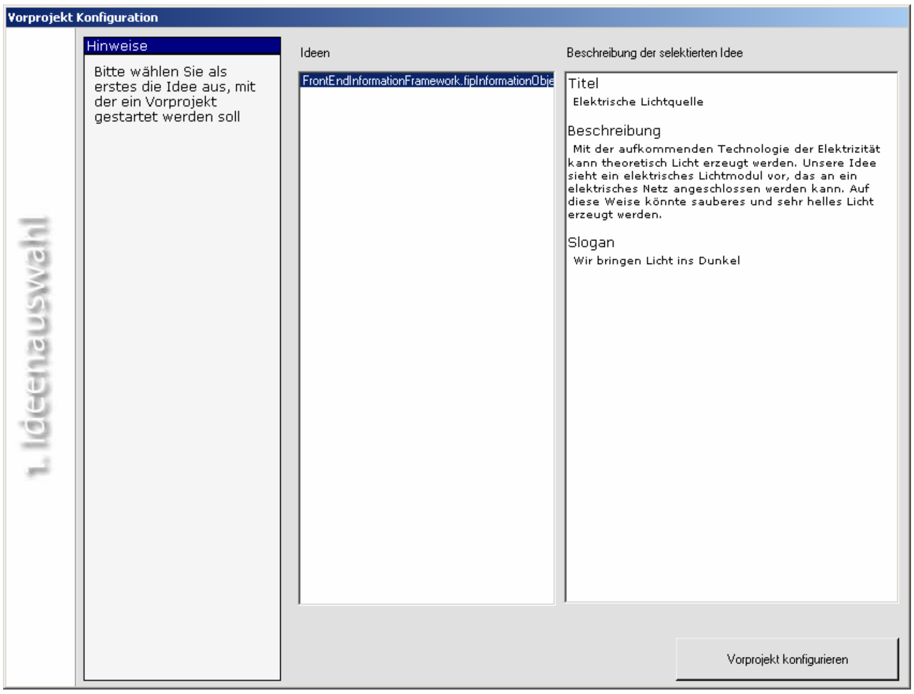

Figure 5. Idea Selection

Starting point to every pre-study is an idea which has been selected to be deepened in a pre-study project. In our software the Innovation Manager can find such ideas in the *Idea Centre* (Figure 5) from where he can select an idea and start the configuration process.

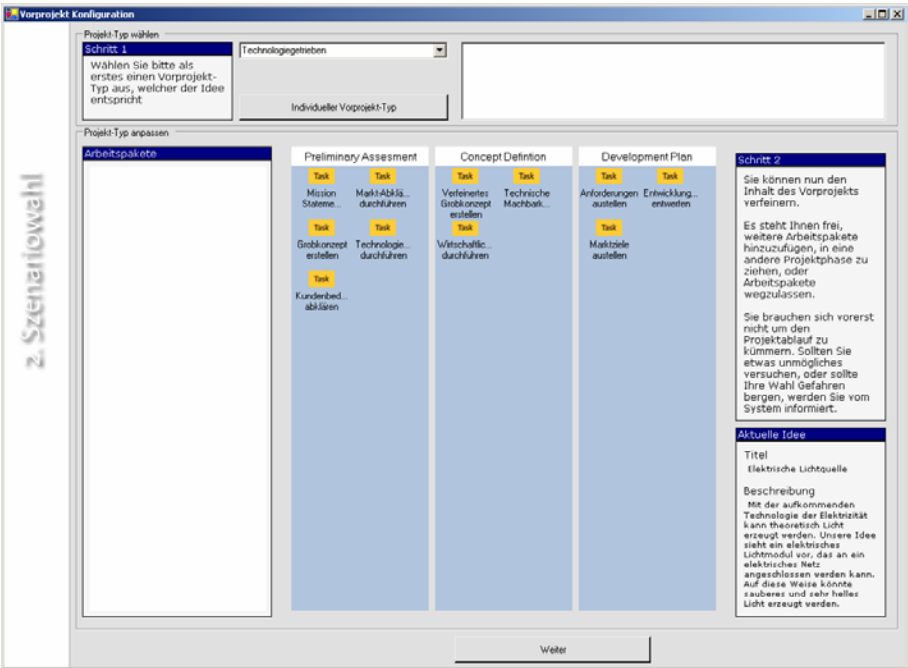

Figure 6. Selection of the Tasks

First step in the configuration process is to classify the idea. In the top section a dropdown list shows all the available process templates, which have names such as “Technology driven idea” or “Market driven idea”. If none of the templates fits to the idea, an empty project can be used. The lower section of the Configuration Wizard shows the current process model with its phases and the tasks in it. The treeview on the left shows all available tasks in its categories. In this section, the model can be manually adjusted or build by dragging tasks from the treeview to the process-phase. Once all tasks needed for the particular project are in place, next step is to bring these tasks into a sequence.

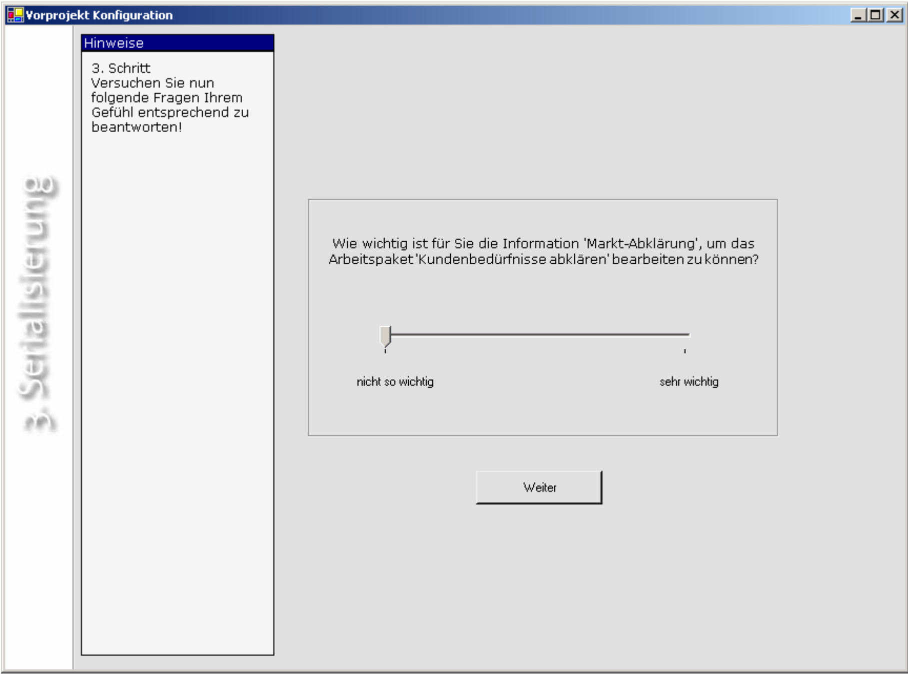

Figure 7. Banding wizard

Figure 7 shows the sequencing wizard. Experiments with the DSM showed that tearing tasks in a matrix of a dimension of 20 tasks already is very hard to do and quite time consuming. Thus, for this tool a new approach for tearing was evolved:

First step is to partition the matrix. If the partitioned matrix has coupled tasks in it a tearing point needs to be found. In literature many different strategies can be found to automatically find such a point by statistical analyses. Our suggestion is to set preference intuitively by asking the responsible Innovation Manager certain questions. The questions are formed of the shape:

“How important is the information” +
[name of an information which is input to the task] + (1)
“for you to execute the task “ + [name of the task]+ “?”

E.g. such a question could be: “How important is the market study for you to execute the task customer analysis?” The answer of the question again depends on the idea we are talking about. Asking such questions importance about the certain information in a particular stage of the process is collected and this information will tell us, where to tear the cycle. This procedure is repeated until no more coupled tasks can be found. Figure 8 schematically gives an overview of the questioning algorithm. After the final partitioning of the DSM we have a set of banded tasks in parallel and sequential order.

These banded tasks can then be exported to a project management software such MS Project™ in our testing environment (Figure 9). In the project management system resources can be assigned to the tasks and a VBA macro helps to automatically generate MS Outlook™ tasks from the project tasks and send them to the responsible person. In MS Outlook™ an other VBA macro allows opening the *Pre-Study Execution* module where the task specific information can be entered.

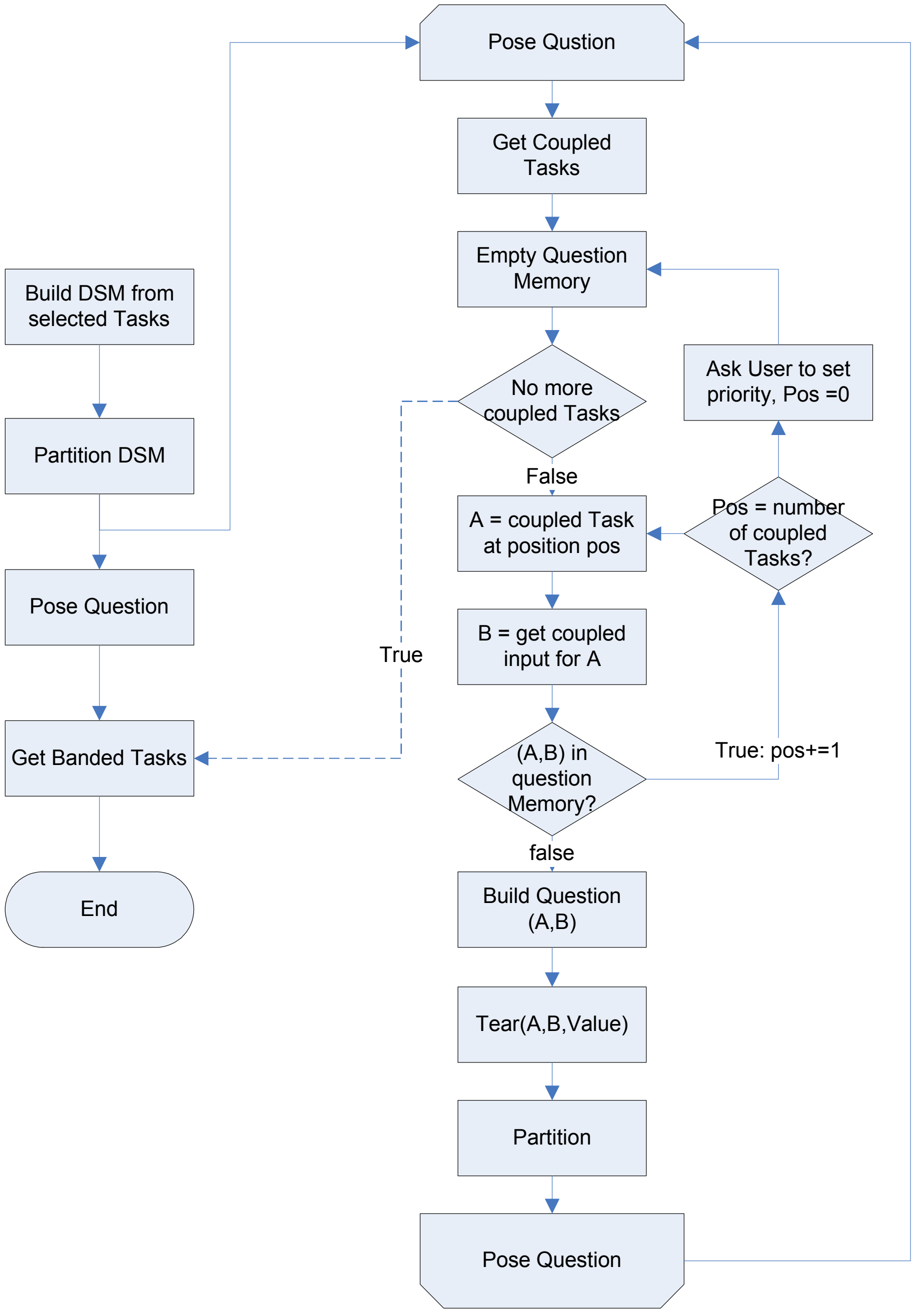

Figure 8. Schema of the questioning algorithm

4.3 Limitations

The solution at its current state is a prototype which acts as a proof of feasibility and has certain limitations. The combination of MS Project and MS Outlook is only a pseudo workflow, although the *Pre-Study Centre* Module allows taking control of the information flow and iterations of tasks.

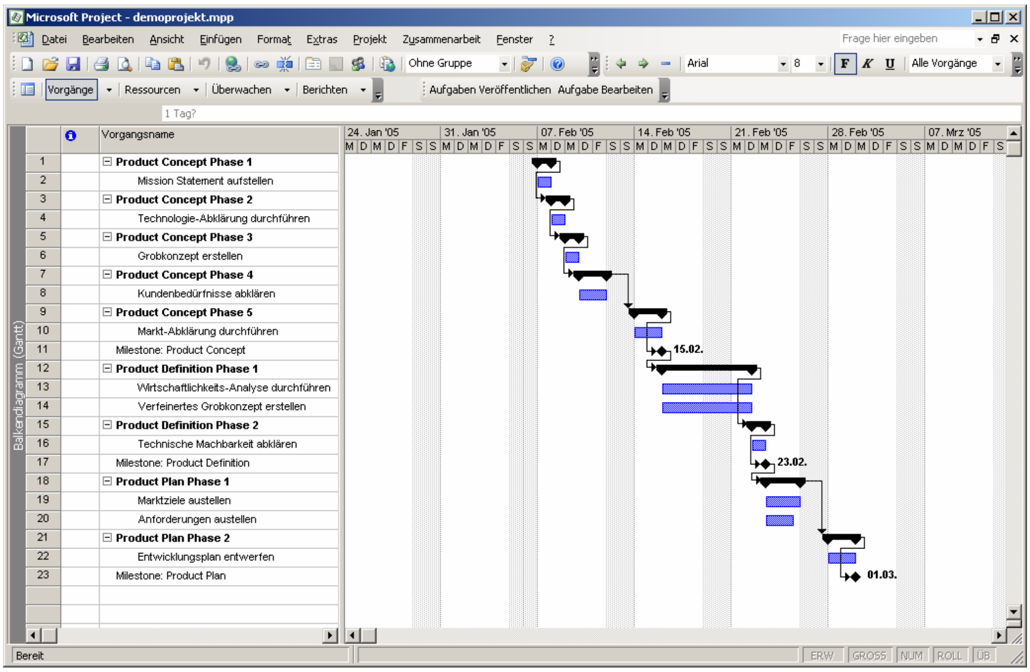

Figure 9. Banded tasks in a project management system

So far, there is only a binary DSM implemented which means that the questioning algorithm only allows "very important" or "not important" answers not a soft weighting of the interdependencies between tasks. Steward suggests in [10] using a numerical DSM with ten different levels of weight for the interdependencies (from "input absolutely required" up to just a "feedback required"). This would facilitate answering the questions and give a more appropriate sequence of the tasks.

Finally, it has to be considered that the DSM sequencing does not pay respect to the availability of resources which will influence the project procedure and its workflow, too.

5 Conclusions and Outlook

The presented Front End framework is able to bring structure into the Front End process and reduce chaotic behaviour in the pre-study phase, but still keeps the process flexible to a high degree. It enables companies to come up with a problem specific tailored process. The developed software shows that the framework is applicable and the definition of a project specific model can be done in a reasonable scope of time. The output of the software can be used to automatically build a project plan.

In general the framework is restricted to the pre-study process and does not deal with the even more unpredictable nature of ideation processes.

Further work has to be done to switch from the binary DSM to a numerical DSM. The current binary dependency levels do not seem give sufficient information about the type of dependency. This would also allow modelling more complex systems with an even more promising result. In addition, effort on optimising the questioning algorithm could reduce the number of questions to be asked during the configuration process.

The output of the configuration would be sufficient information to automatically customise a project specific workflow and a collaborative platform. This would also allow taking control of the iterations during the pre-study process. We aim to investigate the Front End framework in a real world environment to study its practicability in an industrial application.

References

[1] Cooper, R. G. and Kleinschmidt, E. J., "New Products: What Seperates Winners from Losers?" Journal of Product Innovation Management, 4, 1987, 169-184.

[2] Herstatt, C., "Theorie und Praxis der frühen Phasen des Innovationsprozesses", io Management, 1999, 1999, 80-91.

[3] Khurana, A. and Rosenthal, S. R., "Integrating the Fuzzy Front End of New Product Development", Sloan Management Review, 38, 1997, 103-120.

[4] Reinertsen, D. G., "Taking the Fuzziness out of the Fuzzy Front End", Research Technology Management, 42, 1999, 25-31.

[5] Mitterdorfer-Schaad, D. D., "Modellierung unternehmensspezifischer Innovations-Prozessmodelle", ETH Zuerich, Zuerich, 2001.

[6] Nobelius, D. and Trygg, L., "Stop chasing the Front End process -- management of the early phases in product development projects", International Journal of Project Management, 20, 2002, 331-340.

[7] DeVries, V. and Bircher, M., "Integration of influencing factors in the front end of the new product development process", R&D Management Conference, 2004,

[8] Schachtner, K., "Ideenmanagement im Produktinnovationsprozess", Wiesbaden, 2001.

[9] Hollingsworth, D., "The Workflow Reference Model", Workflow Management Coaliation, 1995.

[10] Steward, D. V., "Re-engineering the design process", Enabling Technologies: Infrastructure for Collaborative Enterprises, 1993. Proceedings., Second Workshop on, 1993, 94-98.

[11] Nyffenegger, F. and Ott, C., "A Software Benchmark on Idea and Pre-Study Management Systems", Centre for Product Development, ETH Zurich, Zurich, 2004.

[12] Paetzold, K., "Workflow-Systeme im Produktentwicklungsprozess", Symposium "Design dor X", Neukirchen, 2004,

[13] Erich Gamma, R. H., Ralph E. Johnson, "Design Patterns", Addison-Wesley Professional, 1997.

[14] Prepall, E. and Arsal, K., "Development of a Front-End Project Management Tool for the Product Innovation Process", Center for Product Development, ETH Zuerich, Zuerich, 2004.

[15] Kuehn, A., "Systematik des Ideenmanagements in Produktentstehungsprozessen", Fakultät für Maschinenbau, Universität Paderborn, Paderborn, 2003.

[16] Assaf Arkin, I., "Business Process Modeling Language", BPMI.org, 2002.

[17] Unknown, "Workflow Process Definition Interface -XML Process Definition Language", Workflow Management Coalition, 2002.

[18] Kusiak, A., Nick Larson, T. and Wang, J. R., "Reengineering of design and manufacturing processes", Computers & Industrial Engineering, 26, 1994, 521-536.

Felix Nyffenegger, Nadja Jamali, Dr. Carmen Kobe, Prof. Dr. Markus Meier
Federal Institute of Technology, Centre for Product Design, CH-8092 Zurich, Switzerland
Phone: +41 44 632 47 61, Fax: +41 44 632 11 81, E-mail: nyffenegger@imes.mavt.ethz.ch