INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

SOFTWARE WIZARD DESIGN FOR COMPLEXITY MANAGEMENT APPLICATION

M. Parvan, M. Maurer and U. Lindemann

Keywords: software wizard, assistance system, user interface, complexity management, multiple domain matrix

1. Situation

Many software systems are characterized by a continuous increase of internal and external complexity and are affected by conflicting requirements regarding costs, time and quality. Different types of assistance systems are in use for facilitating the handling of complex software. These systems vary by criteria such as the specific application environment, utilized tools, user type, software system functionality etc. Hereby the addressed users play an important role. Various types of users – experienced, non-experienced, with more or less software and methodology application know-how – need different types of assistance. Consequently, the assistance system has to be tailored for specific user needs.

In general, the design of a new software system, and in this case the design of a application system, has to follow the requirements of all addresed users. However, complex application systems (e.g. word processors, spreadsheets, database applications etc.) can only be handled by expert users. This situation generates conflicts between highly complex and powerful application systems on one hand and the user requirements regarding easy handling of functions on the other. The implementation of a suitable assistance to guide the user through the software application represents the consequence of the displayed conflict.

The design of assistance systems, so called “wizards”, which support the analysis, interpretation and handling of complex systems, is the research subject of this paper. The wizards are customized to the specific design and content of a particular application system. The system will be described in the following. The research question stated concentrates on the design methods, processes and their combinations needed to develop a useful and effective user assistance.

The complex application system LOOMEO, is considered here. LOOMEO was developed by TESEON GmbH in cooperation with the Instiute of Product Development of the TU Munich. LOOMEO is a tool for dynamic visualization, analysis and optimization of system structures. The LOOMEO functionalities are based on complexity management methods and their application on networks of system dependencies. This application requires a high algorithmic assistance (for more details regarding LOOMEO see www.teseon.de).

The design of wizards for the displayed software system has to fulfil various objectives:

- Optimization of software handling
- Method-specific assistance for users
- Optimization of information acquisition (e.g. by particular user interfaces), problem identification and identification and interpretation of analysis results
- Optimized provision of alternatives
- Generic description and classification of designed wizards for future implementation

2. Motivation and Requirements for Complexity Management Wizard Design

The LOOMEO software tool was generated for efficient visualization, analysis and optimization of complex systems. By using LOOMEO the system becomes transparent regarding its structure and behaviour. Dependencies and interactions between elements can be visualized and processed with matrix and graph theory based methods. Thus, with LOOMEO the complexity based on large numbers of dependencies can be managed; and the effort for analysis and visualization can be reduced considerably. However, LOOMEO is a highly complex system and can only be applied effectively by experienced and skilled users so far. In order to make LOOMEO applicable for non-experts, an easy handling and application of the software must be realized. This can be implemented by using wizards; the users then can benefit from many advantages like fast generation of results and decision making.

2.1 Wizard Definition and Classification

A wizard is defined as an interactive utility that guides users through a multi-step, infrequently performed task. Effective wizards reduce the knowledge required to perform the task compared to alternative user interfaces [Microsoft 2009].

Compared to a simple help function a wizard communicates directly with the user and offers him work steps which can be performed automatically. A wizard is therefore a computer-based assistance system, which supports the user in executing a wide range of different actions.

According to [Rech et al. 2006] the main goal of intelligent assistance consists of the enabling and optimization of:

- The automation of simple or self-repeating software design tasks like compilation, creation of test cases, identification of code etc.
- The insight in designed system over cross references, inquiry possibilities and visualization.
- The interaction with and negotiation between participants, e.g. users and alternatively assistance sub-systems, for supporting cooperative work or explaining the performed assistance.

From a user perspective, wizards have to be easy and intuitive to handle and have a modular construction. Thus wizards can be completed by further modules and so extended in their functions. The modular construction enables in addition an efficient and central maintenance of the wizards.

The approach of wizard design can be illustrated by the depiction of the assistance system data flow (see Figure 1). Information about the user, the work content (existing documents), the history, the current conducted activities (processes) and the tool status are being collected in a widespread data pool. Hence information can be used to design an assistance (wizard) which than subsequently can be offered to the user in a specific form [Rech et al. 2006].

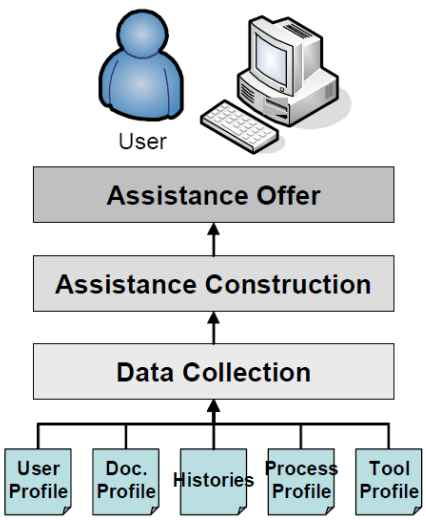

Figure 1. Assistance system data flow [Rech et al. 2006]

Wizards can be classified by many different criteria. The software and therefore wizard user addressed plays an important role in wizard design. Hence a user-centered classification is appropriate for this specific situation. Such a classification can be realized by allocating existing assistance systems to the six main stages of human action (see Figure 2).

Motivation, activation and goal setting

- Activation assistance
- Coach assistance
- Warning assistance
- Orientation assistance

Perception

- Display assistance
- Amplification assistance
- Redundancy assistance
- Presentation assistance

Information integration / Situation awareness

- Labelling assistance
- Interpreter assistance
- Explanation assistance

Decision-making, action selection and action execution

- Supply assistance
- Filter assistance
- Adviser assistance
- Delegation assistance
- Take-over assistance
- Informative execution assistance
- Silent execution assistance

Action execution

- Power assistance
- Limit assistance
- Dosing assistance
- Shortcut assistance
- Input assistance

Processing feedback of action results

- Feedback assistance
- Critique assistance

Figure 2. Classification of assistance systems by human action stages [Wandke 2005]

Many wizards that help users to install or to execute a computer program belong to the category of filter assistance. When for example only one option is presented they belong to the next category: adviser assistance [Wandke 2005]. Here the designed wizards highlight characteristics of many of the displayed assistance systems (see Figure 2). The wizards comprise characteristics for example from the adviser, delegation, supply, filter, silent execution and many more of the classified assistance systems.

2.2 Wizard Application and Usability Design

The wizards described in this paper display a method-specific design. The methods applied consist mainly of complexity management methods and principles. In LOOMEO mainly methods for variance, change and process management and the principles transparency and modularization are embedded. In addition, an extension of LOOMEO by the FMEA-based risk management method is planned for the future and therefore must be integrated in the wizard development plan. The displayed methods and principles represent the input for the development of the complexity management wizards.

To avoid problems that can occur due to the integration of all displayed methods and principles in one universal wizard, several wizards have to be designed. In fact, for each method and principle integrated in LOOMEO – for every application area – an independent wizard has to be developed. Finally, the following six wizards were generated: transparency wizard, modularization wizard, variant management wizard, change management wizard, process management wizard and FMEA-based risk management wizard.

Wizards are used for the direct interaction with users and represent software ergonomic tools which play an important role in man-machine systems. Without wizard support and guidance a sound communication between man and machine often seems to be impossible. The software ergonomist Jakob Nielsen developed ten general principles, called usability heuristics, to support the design of wizards and application systems in general [Nielsen 1993]: simple and natural dialogue, speak the user's language, minimize user memory load, consistency, feedback, clearly marked exits, shortcuts, good error messages, prevent errors and help and documentation.

The wizards described in this paper were generated considering the fulfillment of the discussed requirements, classifications and usability heuristics.

3. Realisation and Results

Wizard design is a highly demanding process, because of the high amount of specific factors involved. The most important influence factors are:

- **The software:** The used software represents the main influence factor on wizard design. The wizard accesses, depending on the considered problem, different functionalities and visualization alternatives of the software and offers them to the user in an organized and structured form.
- **The problem:** The application purpose can be deducted from the considered problem. Different wizards have to be applied depending on the problem that has to be solved. For this reason the considered problem determines the entire design of the wizard and specifies how extensive and in which form the assistance has to occur.
- **The method:** The software-integrated methods affect mainly the type and range of embedded wizard-functionalities. The method specifies the application area of the wizard.
- **The programming language:** The applied programming language affects the wizard design. Especially the layout and design of the user interface is of major importance.

The six mentioned wizards exhibit various characteristics (and differ more or less from each other), as a result of the possible combinations of the described influence factors.

The transparency wizard comprises all visualization and navigation functionalities and supports the transparent and structured design of the system. With the aid of this wizard complex systems can be analyzed from a qualitative point of view. The modularization wizard enables the generation of modules and repeat parts. The variant management wizard supports the reduction and control of variance inside of complex technical systems. The change management wizard enables the tracing and optimization of realized changes. The process management wizard supports the control and optimization of activities and structural characteristics of complex processes. Finally the FMEA-based risk management wizard enables the reduction and control of occurred risks.

For the creation of the six wizards four main work steps had to be implemented:

1. Project data acquisition
2. Data structuring, classification and processing

3. Wizard verification and validation through workshops
4. Revision and conversion of workshop results

Eight projects served as data basis for the data acquisition. The projects tackle real industry problems and were conducted by the TESEON GmbH team. The projects differ considerably from each other regarding methodology, objective and functionality; the projects were chosen as representatives of the six identified complexity management application areas. If a wizard had for example been used at the time of processing the project in which modules were generated to enable a fast and cost efficient assembly of technical products, the duration time of the project could have been shortened substantially and more solutions could have been identified.

The selected project data had to be structured on the basis of a framework to enable a systematic and goal-oriented wizard design process. Within the selected projects six standard process steps could be identified, which are used for the classification of the selected project data [Lindemann et al. 2009, Maurer 2007]:

1. Problem definition
2. System definition
3. Information acquisition
4. Deduction of indirect dependencies
5. Structure analysis
6. Product design application

After the comprehensive scientific analysis the necessity of a modification of the six classic standard process steps turned out to be necessary. The new adjusted standard process steps are used for the final structuring of the project data:

1. Template application
2. Problem definition
3. System definition
4. Information acquisition
5. Analysis strategy definition
6. Analysis

The new standard process steps were defined during the execution of several workshops.

3.1 Wizard Design

On a highly abstract level, the general design is similar for all six presented software wizards (see Figure 3). The wizard receives from the software user information regarding the considered problem, the system to be analyzed and the existing constraints.

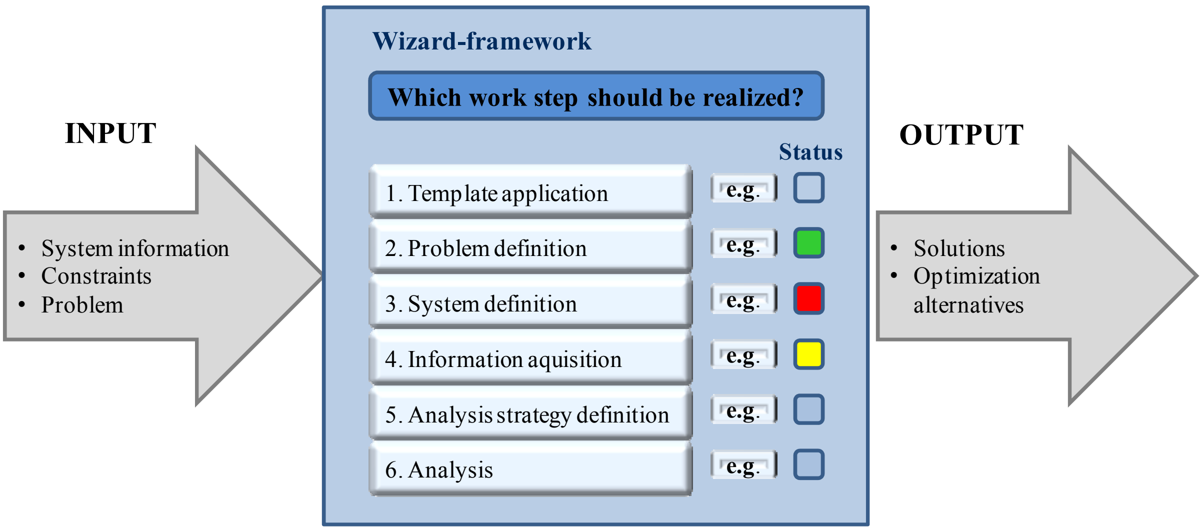

Figure 3. User interface framework

Next, the problem is processed inside of the method-specific wizard. The wizards consist of a framework which contains a variable number of standard process steps. These process steps can be partially executed in an arbitrary order; and in most cases they must not all to be run through mandatorily. After processing the wizard one or more problem solutions and optimization alternatives emerge and are offered in a structured form to the user.
The general wizard functionality is characterized by its various processing paths, which the user receives as selection alternatives. The multitude of processing sequences results from the various arrangement alternatives of integrated instructions. The sequences differ from each other regarding the type and number of contained instructions and regarding the form of giving advice, displayed examples and required tools. The overall representation of the wizard functionality as a flowchart with all possible processing paths is shown in Figure 4.

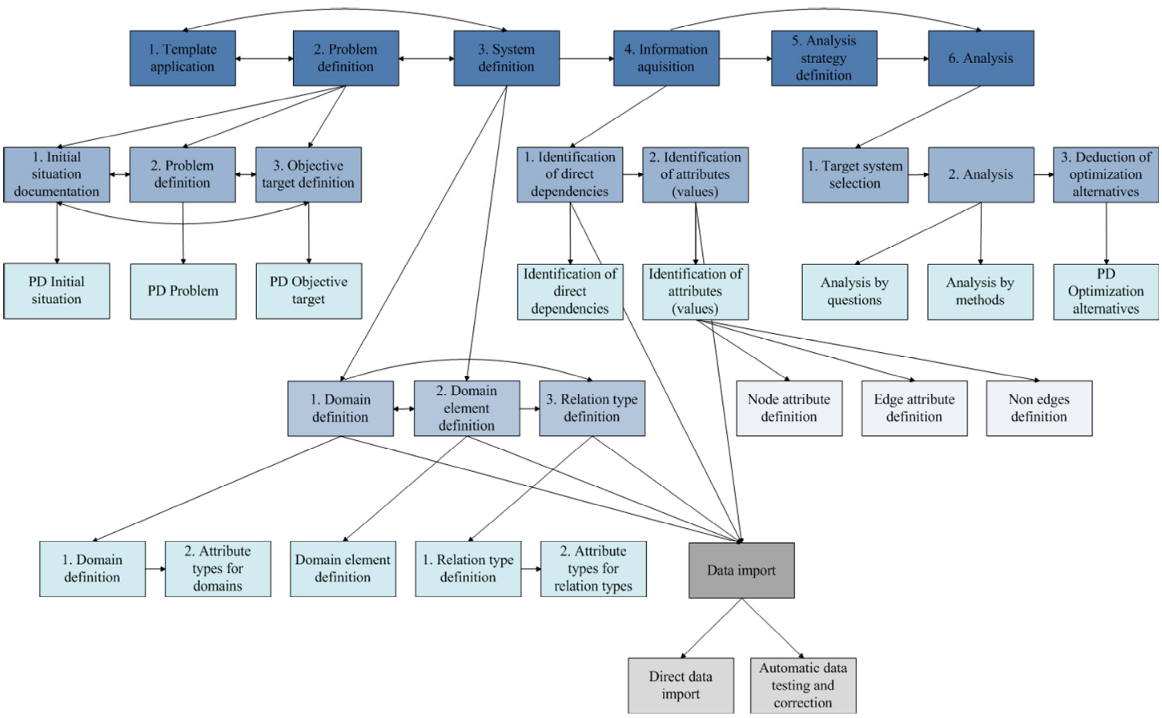

Figure 4. Wizard functionality flowchart

3.2 Wizard Characteristics and Application

The wizard characteristics are related to the described principles and methods of complexity management presented in Chapter 2. For each principle and method an independent wizard had to be generated. The six designed wizards – the transparency, the modularization, the variant management, the change management, the process management and the FMEA-based risk management wizard – are described in the following regarding their particular design and functionality. Hereby, only the most representative functionalities, work steps and analysis alternatives are going to be illustrated. In addition some of the differences of the wizards are highlighted.
The transparency wizard possesses a generic specification and contains all six new standard process steps (see Chapter 3). The design of the analysis process step represents the most interesting aspect. The analysis of system transparency is, compared to the other five wizards, the only standard process step that contains two analysis alternatives. The analysis of system transparency can be conducted by questions or with the aid of methods. Examples for contained analysis questions are:

- What is the cause of problem appearance for a system element?
- From which other element a system element is influenced?
- How important is a particular element?
- …

The offered analysis methods are designed as "mini-wizards" and the user must follow a range of preset instructions for system analysis and result generation.
The modularization wizard contains a reduced number of standard process steps. By the application of templates some of the work steps become dispensable and can therefore be removed from the wizard structure. For this reason, the modularization wizard contains only five of the six standard process steps and a reduced number of sub-steps. Figure 5 shows a typical template for system definition from the modularization wizard. The displayed domains and relation types cannot be changed or adjusted by the software user.

	Components (C)	Functions (Fct)	Features (F)
Components (C)	C-change can induce C-change (Geometry and Cross-section)	C serves for Fct	C can influence F
Functions (Fct)	–	Fct needs Fct (Functional modeling)	–
Features (F)	–	–	F can influence F

Figure 5. Multiple Domain Matrix (MDM) template with recorded domains and relation types

The variant management wizard reduces the number of required process steps and highlights method-specific templates. The variant management wizard contains only five of six standard process steps.
An important characteristic of the variant management wizard is the possibility to use a variance report as a visualization tool. The variance report represents a particular visualization of the variance existing inside of a complex system (see Figure 6.). Within the variance report the results of all conducted variance analysis are combined in one view. Through the application of the variance report the user can identify and evaluate all results of realized variance analysis at a glance. Further the user can conduct system improvements on the basis of the received results.

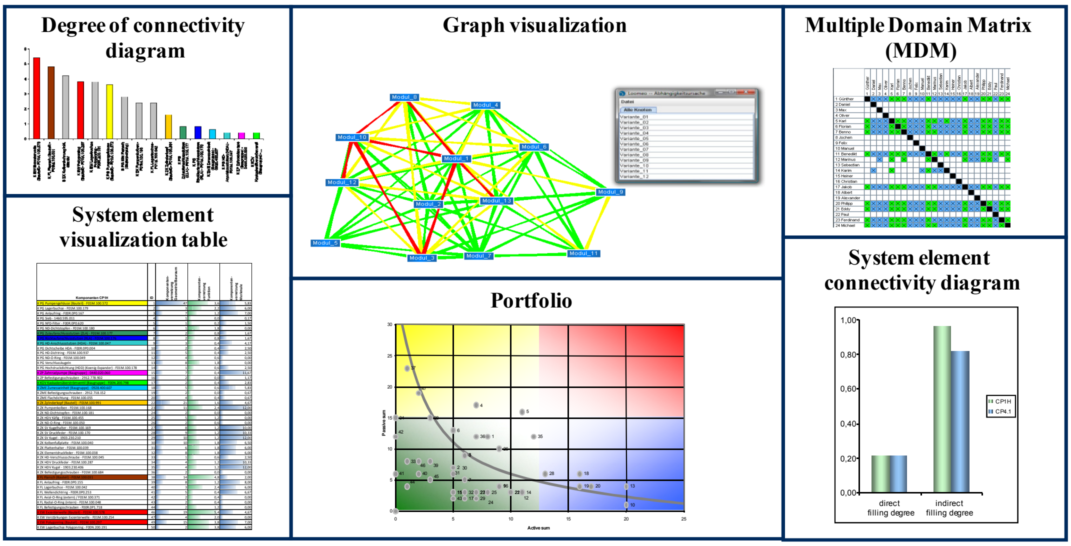

Figure 6. Variance report

The change management wizard comprises all six standard process steps. Thus, the user obtains more freedom regarding system visualization and analysis.

Hereby, the analysis of change is conducted by predefined questions. Examples for change analysis questions are:

- Which influence results from the change of a system element?
- Which change paths exist between two system elements?
- Which system elements generate minor changes?
- …

The user chooses one of the displayed analysis questions and obtains the allocated mini-wizard of the analysis method. Examples for contained change analysis methods are:

- Impact analysis
- Significant element structures (explore, path, cluster analysis etc.)
- Graph representation
- …

The process management wizard cannot be defined as detailed as the other wizards. Some of the standard process steps could only be described schematically because of the low number of projects analyzed in the research area of process management and the absence of a specific approach and methodology for the realization of process management by using LOOMEO.

A specific characteristic of the process management wizard is the visualization and analysis of a system with the aid of entity relationship (ER) modeling. The possibility to display complex processes and systems as ER-models will be implemented in LOOMEO in the future as an additional functionality and will enable the user to handle the observed system more effective and realize an overall analysis of complex processes.

Currently only a few ideas exist regarding the realization of the FMEA-based risk management wizard. What is already settled is that the risk management wizard will contain all three FMEA types: System-, Product- and Process-FMEA. In LOOMEO the FMEA process will be realized inside of a Multiple Domain Matrix (MDM) or alternatively by use of a dynamic graph. Figure 7 shows the design of a MDM template with recorded domains for FMEA realization.

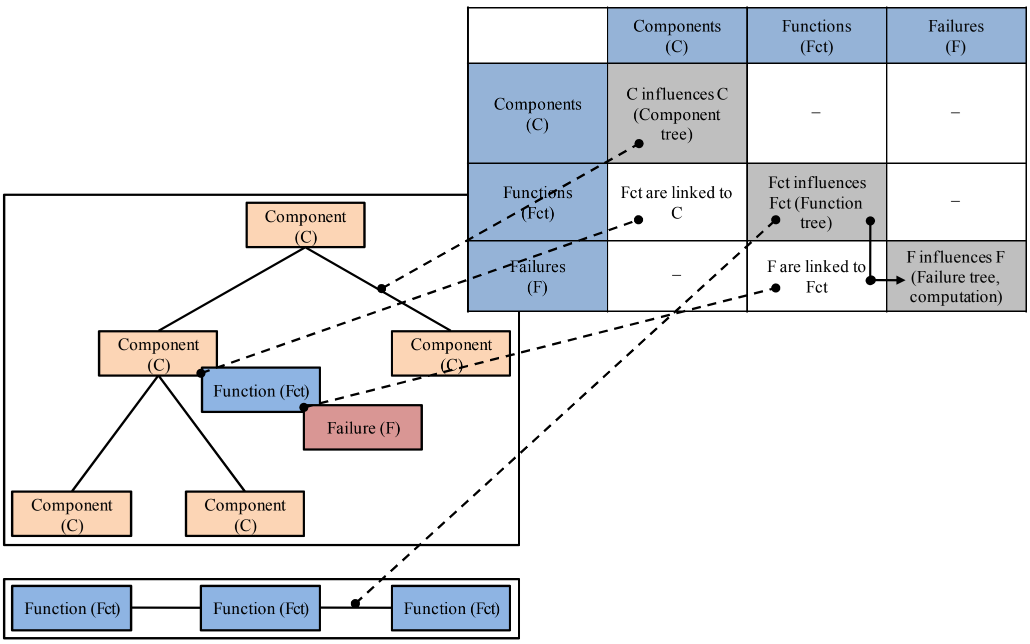

Figure 7. Multiple Domain Matrix (MDM) template with recorded domains

Due to the existing dependencies between functions and failures, failures caused by inserting a new component into the system can be identified by using the matrix displayed in Figure 7. Further, by the observation and analysis of function cycles, failure cascades may also be identified. The function tree must be generated manually. The failure tree results from the function tree and the dependencies between failures and functions by calculation.The FMEA-based risk management wizard is designed for the evaluation and control of risk in complex systems and its main objective is the optimization of technical systems concerning the contained failures and their cause, type and consequence.

4. Discussion

The realization and results of wizard design highlight many differences regarding content, completeness and analysis scope. These differences result mainly from the underlying wizard methodology and from the existing experience and knowledge base. Depending on the information range existing at this point, the wizards can be more or less defined in detail.

The five wizards, which have been designed in detail – the transparency, the modularization, the variant management, the change management and the process management wizard – differ from each other regarding various criteria. On the basis of these criteria the wizards can be evaluated and compared with each other (see Figure 8). The evaluation criteria “Further development demand” describes the degree of wizard development at the actual stage and integrates aspects like designed frameworks, completeness of integrated analysis tools, level of method integration and adaptation etc. The evaluation pattern is realized by symbols (“++” for very high, “+” for high, “0” for medium, “-“ for low and “ - -“ for very low).

Wizards	Evaluation criteria					
	Specialization	Functional range	Freedom of decision	Application of predesigned templates	Work step range	Further development demand
Transparency wizard	– –	++	++	– –	++	–
Modularization wizard	++	– –	– –	++	–	–
Variant management wizard	+	0	0	+	0	0
Change management wizard	0	+	+	–	+	0
Process management wizard	+	0	0	+	0	+

Figure 8. Wizard evaluation

The transparency wizard exhibits the lowest specialization and therefore the highest functional range, freedom of decision and the highest amount of existing work steps. The application of existing templates plays a rather unimportant role and serves only as an additional information acquisition method. The transparency wizard is precisely defined and therefore a further development is not required.

The design of the change management wizard is similar to the design of the transparency wizard. The change management wizard displays however a higher specialization, because of the defined target course of change tracking and control, a slightly lower functional range, a high freedom of decision and the same range of existing work steps. The slightly higher usage of predesigned templates is closely connected to the higher specialization. The change management wizard displays in addition a medium development demand.

The variant and the process management wizard are relatively similar in design. Compared to the transparency and the change management wizard both wizards display a higher specialization and therefore a lower functional range, a lower freedom of decision and range of existing work steps and a higher usage of predesigned templates. The demand for further development is for the process management wizard the highest due to the lack of experience and realized projects in this research area. The demand for further development can be situated for the variant management wizard on the same level as for the change management wizard.

The modularization wizard represents the “antipole” of the transparency wizard. The modularization wizard highlights the highest specialization of all wizards and therefore the lowest functional range, freedom of decision and range of existing work steps and the highest application of predesigned

templates. The entire objective target of the modularization wizard concentrates on the adaptation and application of predesigned templates. Furthermore, the modularization wizard is precisely defined and can be situated on the same development level as the transparency wizard. The requirement for further development is, due to the detailed design of the modularization wizard, rather low.
The objectives set at the beginning of the wizard design project are almost completely fulfilled. For the five more detailed defined wizards the optimization of the LOOMEO software handling could be achieved. Further the realized assistance highlights a precise method specific design. A high optimization of information acquisition, by using particular designed user interfaces and predesigned templates, and a faster and more effective problem identification and identification and interpretation of analysis results could also be realized. The offering of several results, solutions and optimization alternatives was the most important objective of this project and could be completely fulfilled. Finally an overall theoretical description and classification of the designed wizards for a future implementation was realized.

5. Conclusion and Outlook

The common goal of the designed software wizards represents the reduction and management of the inner and outer complexity of a technical system from many different points of view. The six wizards highlight different characteristic and a more or less detailed method-specific design. The software user can apply (depending on the observed system and the occurred problem) one or more wizards to achieve solutions and optimization alternatives. The five detailed designed wizards are based on data extracted from real projects and highlight the present information level. The schematic designed FMEA-based risk management wizard illustrates the desired enhancement of the LOOMEO software.
The transparency wizard, mainly used as a tool for system visualization and navigation, is precisely defined and must be completed only regarding the integrated functionalities. The modularization wizard is also precisely defined and comprises even customized user interfaces, which have to be further adjusted. For the variant management wizard an extension of the implemented functional range as well as an optimization and adaptation of developed tools and user interfaces can be expected for the future. The further development of the change management wizard aims at the extension of the embedded system analysis and information acquisition functionalities as well as on the optimization of the integrated work steps. The design of the risk management wizard exists in present only in form of ideas regarding the methodical content and the possible visualization alternatives and represents therefore the at least defined wizard.
The integrated functionalities and design characteristics can be extended and optimized by conducting further projects in the six displayed research areas and the implementation of method-specific features and tools inside of the LOOMEO software. The LOOMEO software can be easely utilized in the future by all types of users for the reduction and control of the overall system complexity with the aid of method-specific designed wizards.

References

Rech, J., Ras, E., Decker, B., "Intelligente Assistenz in der Softwareentwicklung 2006: Zusammenfassung der Ergebnisse", Frauenhofer IESE, Kaiserslautern, 2006.

Lindemann, U., Maurer, M., Braun, T., „Structural Complexity Management", Springer Verlag Berlin, 2009.

Nielsen, J., „Usability Engineering", Academic Press, San Diego, 1993.

Wandke, H., "Assistance in human-machine interaction: a conceptual framework and a proposal for a taxonomy", Theoretical Issues in Ergonomics Science, Vol. 6, No. 2, March-April 2005, p. 129-155.

Maurer, M., "Structural Awareness in Complex Product Design", Dr. Hut Munich, 2007.

Dipl.-Ing.Manuela Parvan
Institute of Product Developement, Technical University Munich
Boltzmannstrasse 15, 85748 Munich, Germany
Telephone: +49 89 289 15137
Telefax: +49 89 289 15144
Email: manuela.parvan@pe.mw.tum.de