Semantic Execution of Subject-Oriented Process Models

Albert Fleischmann[1], Werner Schmidt[2], and Christian Stary[3]

[1] Metasonic AG, Münchner Str. 29, D-85276 Pfaffenhofen, Germany
Albert.Fleischmann@metasonic.de
[2] University of Applied Sciences Ingolstadt, Esplanade 10, D-85049 Ingolstadt, Germany
Werner.Schmidt@haw-ingolstadt.de
[3] University of Linz, Freistädterstraße 315, A-4040 Linz, Austria
Christian.Stary@jku.at

Abstract. Workflow Management Systems (WFMS) are becoming increasingly important as tools to support people involved in the execution of business processes and to automate parts of it. As business processes involve several actors with varying backgrounds, workflow engines need to offer appropriate interfaces in order to be accepted and deliver the expected benefits. In this paper we present a structural interface design based on general user interface requirements and special properties of workflow systems, in particular of a subject-oriented workflow engine.

Keywords: Business process management, workflow systems, user interface, structural design, Subject-oriented BPM.

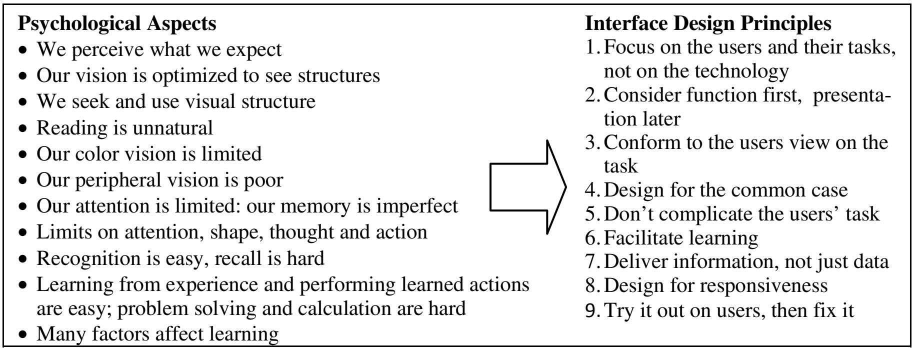
1 Introduction

Workflow Management Systems (WFMS) support the management of business processes both at design and at runtime. A modeling component allows specifying the process, while a Workflow or Process Engine (WE, PE) controls the execution of process instances according to the model. The WE navigates users through the steps of a process they are involved in and might integrate IT applications to accomplish process-related tasks. WFMS require a well-designed user interface (UI) as a critical factor for their success. Interface design is no longer considered to be art, but 'a kind of joint computer-cognitive engineering, that is, science-based techniques to create interactive systems satisfying specified requirements' (see Card's foreword in [6]). Consequently, a user interface needs to meet specified requirements in order to be accepted by its users. In section 2 we outline general guidelines for designing user interfaces and then look at domain-specific requirements for workflow engines (section3). Based on the requirements in section 4.2 the design proposal is worked out, recognizing the properties of Subject-oriented Business Process Management (S-BPM) (section 4.1). The design is validated and modified according to the evaluation results (sections 4.3 and 4.4). The contribution concludes in section 5.

M. Kurosu (Ed.): Human-Computer Interaction, Part I, HCII 2013, LNCS 8004, pp. 330–339, 2013.

2 General Design Guidelines for User Interfaces

There are many publications available on guidelines for user interfaces and websites (e.g., see [1] [6] [7] [9] [12] [13]). They do not precisely describe design activities, but define goals. The guidelines do not give special recommendations for particular application classes like WFMS, and they are ‘quite similar if we ignore differences in wording, emphasis and the state of computer technology when each set was written’ [6]. In his book ‘Designing with the mind in mind’ Johnson describes the most important aspects of psychology underlying user interface and usability guidelines. Figure 1 shows them together with the corresponding design principles [5] [6].



Psychological Aspects

- We perceive what we expect
- Our vision is optimized to see structures
- We seek and use visual structure
- Reading is unnatural
- Our color vision is limited
- Our peripheral vision is poor
- Our attention is limited: our memory is imperfect
- Limits on attention, shape, thought and action
- Recognition is easy, recall is hard
- Learning from experience and performing learned actions are easy; problem solving and calculation are hard
- Many factors affect learning

Interface Design Principles

1. Focus on the users and their tasks, not on the technology
2. Consider function first, presentation later
3. Conform to the users view on the task
4. Design for the common case
5. Don’t complicate the users’ task
6. Facilitate learning
7. Deliver information, not just data
8. Design for responsiveness
9. Try it out on users, then fix it

Fig. 1. Psychological aspects and design principles for user interfaces

The principles in the figure are mainly of static nature. They do not refer to operational aspects relevant for interfaces in use. Nievergelt proposes to consider these aspects by coining the following questions that should be answered for each situation in interaction: Where did I come from? Where am I? Where can I go from here? [11]. Users of workflow engines when executing sequences of tasks need orientation and navigation support to know where they are, where they have been, and where they can go. Although support for visualizing navigation structures is essential (cf. [9]), many applications lack, e.g., the ‘you are here’ indication. When Nievergelt’s design issues are tackled several concepts facilitate answering the questions: Trails refer to past actions (i.e. orientation w.r.t. to the past), sites correspond to the current action or information to give (i.e. orientation w.r.t. to the current situation), and modes are about possible actions to come (i.e. orientation w.r.t. to future activities). Trails have also been called feedback, sites have been called responses, and modes have been called openings (‘What can you do?’). When users carry out more than a single task when communicating with interactive systems, so-called field tasks, reached thanks to the responses, have been distinguished from so-called interaction tasks including feedbacks and openings [3]. Hence, trails, sites and modes reflect interaction patterns with certain meanings that correspond to handling workflows interactively. Workflow systems may engage users in several tasks in a certain period of time requiring trails, sites, and modes to act in line with active business processes. We will refer to trail (Now), site (Needed), and modes (Next) in our design approach - see section 4.

3 User Interface Requirements for Process Engines

As we are talking about user interfaces the focus is on how workflow engines integrate humans in the execution of instances of various process types with participants diverse in hierarchical position, education, computer literacy etc. (from CEO to blue collar worker). The PE should reduce cognitive overload by providing an intuitive and easy to use interface (process portal), framing both its own functionality to execute process instances and the embedded applications with their particular user interfaces. The UI requirements can be derived from the functions a WE needs to provide for actors in processes. For illustrating those, the typical work at a conventional office desk can serve as a metaphor. The desk is equipped with many tools helpful for all types of processes like personal computer, notepad, inhouse-mail envelopes, in-tray, out-tray, stapler etc. There are also tools which are specific for a certain process like forms for purchase orders or vacation requests. When working on process instances at the desk a person for example takes inhouse-mail envelopes out of the in-tray, opens them, selects one (e.g. price calculation request) and starts activities necessary to accomplish the tasks related to the case. This could mean calculating a price with a spreadsheet software on the PC, fill the result in the request form, put it in an envelope, add the addressee and put it into the out-box. Another typical situation might be the person itself instantiating a process by filling in a vacation request form and sending it via inhouse-mail to the responsible manager for approval. In addition to supporting those activities, a workflow engine, due to its overarching of single work places, can also deliver functions like status reporting etc. The following list contains major functions which set the requirements for a user interface properly presenting them.

- **Instantiating processes and tracking.** A user needs to be able to select and start processes he/she is allowed to initiate (according to organizational settings). He should be able to observe the status of instances once they have been started.
- **Receiving and Selecting.** A user requires a quick overview of and an easy access to open instances he needs to work on (work list).
- **Working.** A user needs to be able to accomplish the steps he is responsible for in the process, e.g., directly starting an application system out of the WE user interface when needed in a step. The applications (or single transactions) possible to call are process-specific and therefore need to be embedded and offered according to the context (e.g., a CRM system in a marketing process). Aborting or suspending the instance execution should be possible as well as continuing it later.
- **Sending.** A user needs to be able to easily pass his work results on to the next actor in the process in row. This includes the system's support by determining or suggesting the right addressee(s).
- **Orientation and Navigation.** While processing instances the user should always be able to obtain information about the whole process in terms of which steps already have been finished, which tasks are his/hers and which are the steps still to go afterwards by other actors. The UI should help deciding on and carrying out activities at each execution state by setting the right defaults and providing possible options.

These requirements correspond to Nievergelt's operational aspects (see section 2), as a user wants to know: Where am I now? What do I need to do? What needs to be done next by whom? We term this the Now – Needed - Next (3N) approach.

4 Designing a User Interface for a Subject-Oriented Process Engine

Based on the considerations in section 3 we propose a particularly structured user interface design, tailored for the domain of workflow systems and focused on the runtime part. It should serve as a blue print for interaction design on various devices (mobile and static), allowing up-to-date technologies for implementation. The UI functions for handling a workflow engine depend on the method in which processes are modeled at design time. Functions especially for navigation through a workflow at runtime (process execution) might differ according to the approach used - e.g., BPMN models can include about 160 symbols [14] which might lead to extensive navigation features. To demonstrate the Now-Needed-Next approach we stick to a straightforward BPM technique, namely Subject-oriented BPM. Before giving the design structure, we briefly outline the approach (for details see [2]).

4.1 Properties of Subject-Oriented Business Process Management (S-BPM)

The subject-oriented description of a process starts with the identification of process-specific roles involved, the subjects, and the messages exchanged between them (see fig. 2). When sending messages, required data is transmitted from sender to receiver via simple parameters or more complex business objects if necessary.

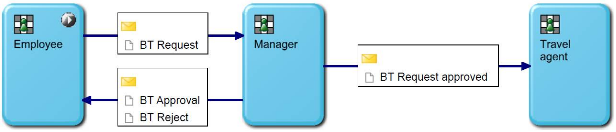


Fig. 2. Interaction structure of the process (BT=Business Trip)

In a refinement step, the modeler describes which activities and interactions the subjects have to perform in which order during process execution, i.e., he defines the behavior of individual subjects. He also specifies business objects as data structures being exchanged with the messages and being manipulated in the subject behavior.

The subject behavior diagram in the left part of figure 3 shows the order in which the employee sends and receives messages, or executes internal actions (functions), and the states he is in during his business trip request process. The initial state is a function state in which the employees complete their business trip request. The state transition 'Fill in BT Request done' leads to a send state in which they send the request to the manager, before entering the receive state, in which the applicants wait for the manager's response. In case they receive a rejection message, the process comes to an end. In case the employees receive the approval message from the manager, they go on the trip on the agreed date and the business trip application process is completed.

The behavior of the manager is complementary to that of the employee (see right part of fig. 3). The manager waits in a receiving state for a request from the employee. After receiving one, he goes to the state of decision, leading either to the approval or rejection. In the second case, a state follows to send the rejection to the employee. In the first case, the manager first moves to a send state for transmitting the approval to the applicant, and then proceeds to a state of informing the travel agent about the approved request. The behavior of the travel office can be described analogously.

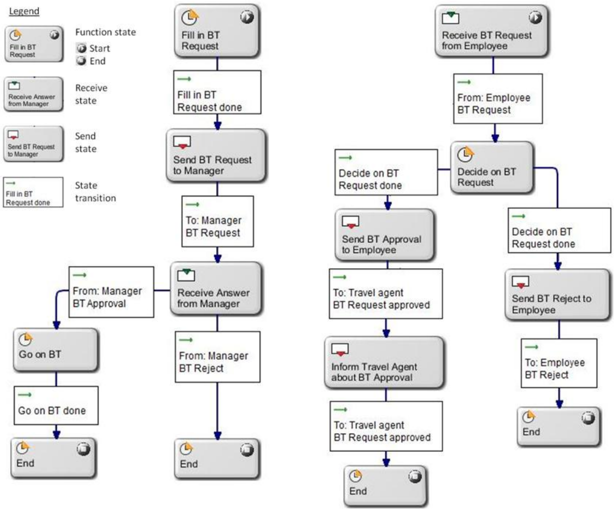

Legend
Fill in BT Request
Function state
Start
End
Receive Answer from Manager
Receive state
Send BT Request to Manager
Send state
Fill in BT Request done
State transition
Fill in BT Request
Fill in BT Request done
Send BT Request to Manager
To: Manager BT Request
Receive Answer from Manager
From: Manager BT Approval
Go on BT
Go on BT done
End
From: Manager BT Reject
End
Receive BT Request from Employee
From: Employee BT Request
Decide on BT Request
Decide on BT Request done
Send BT Approval to Employee
To: Travel agent BT Request approved
Inform Travel Agent about BT Approval
To: Travel agent BT Request approved
End
Decide on BT Request done
Send BT Reject to Employee
To: Employee BT Reject
End

Fig. 3. Behavior of the subjects ‘employee’ (left) and ‘manager’ (right)

Subjects represent active parties in a process as abstract actors. Assigning people to subjects embeds processes in a certain organizational environment. For example all members of a department can execute the behavior of the subject ‘employee’ while the ‘manager’ behavior is reserved for the department head and deputy. Such various embodiments of a business process in an organization are called process contexts.

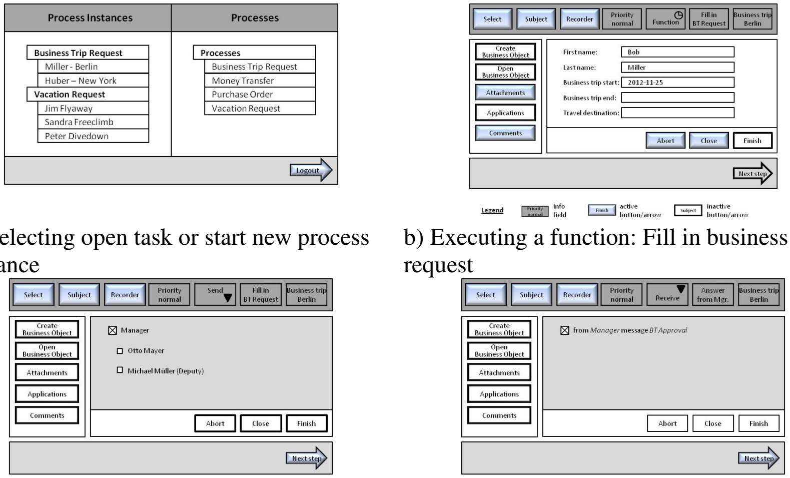
If a business event (e.g., need for visiting a customer in Berlin) in a certain context (e.g., by Bob Miller from the sales department) has to be handled, an instance of the corresponding process (e.g., business trip application) is initiated.

4.2 Structuring User Interfaces for Semantic Execution Support

When structuring a UI for a S-BPM-based workflow system we focus on function and not on graphical aspects (principle 2 in fig. 1). According to the nature of S-BPM only two types of screens are required for execution support: a screen for starting a process instance or selecting existing instances. The second type are screens for the basic workflow operations (do, receive, send). They can be based on one template.

Figure 4a shows the first screen type. The left part lists instances grouped by processes which contain tasks to be accomplished by the user currently logged on (personal work list). In the example he or she has to work on three business trip and two vacation requests (may be as the supervising manager). The right part offers the processes the particular user can start instances of according to the responsibility rules specified in the organization. Selecting one of the four processes listed creates a new instance of the process type chosen, e.g. a request for a certain business trip.

After initiating such a new process instance the user has to execute either a function, a send operation or a receive operation. For that type of operations a common template for the user interface is used. This template (see figures 4b-4d) is based on the Now - Needed - Next (3N) properties described in section 3.



First name: Bob
Last name: Miller
Business trip start: 2012-11-25
Business trip end:
Travel destination:

☒ Manager
□ Otto Mayer
□ Michael Müller (Deputy)

☒ from *Manager* message *BT Approval*

a) Selecting open task or start new process instance

b) Executing a function: Fill in business trip request

c) Selecting a person assigned to subject 'manager'

d) Selecting an expected and available message

Fig. 4. Screen types

- **Now.** The row on top (dark grey background) contains functions to inform the user about the current status of the process instance. They are identical to all screens.
- **Needed.** The middle part (white background) contains all functions which are required for executing internal functions specific to a certain process. They include creating new business objects, opening existing ones or starting an integrated application (arranged on the left). The remainder of the middle part displays the work area in which the user can work on open business objects and finds functions to suspend, abort and finish his activities.
- **Next.** The row on the bottom (light grey background) shows all operations for defining what is coming next.

Figure 4b depicts the screen for executing the function 'Fill in BT Request' as modeled in the behavior diagram for the subject the user represents (here employee, see fig. 3). This information is presented on top together with the priority and name of the instance. Pushing the button 'Subject' displays the entire behavior model of the subject with the current execution state being highlighted. 'Recorder' activates a feature of the process engine, showing the steps already taken in the instance by all subjects involved in the process. ‘Select’ brings the user back to his work list (fig. 4a).

The business object modeled as belonging to the current function is already open as a form. The user needs to type in the required information. The business object form is part of the process specific aspects of the UI. Filling it in can be aborted without saving the inputs (‘Abort’) or suspended with storing the data put in so far (‘Close’). Once all required fields are filled finishing is possible (‘Finish’), activating the functions in the bottom row (here ‘Next step’). By clicking the ‘Next step’ button the workflow proceeds to the subsequent action. In our example this is a send operation which transfers the application form to the manager.

Figure 4c shows the UI for such a send state in which the person or organization assigned to the subject ‘manager’ needs to be known (process context). In our context two people are assigned to the subject and therefore can be offered as addressees by the workflow engine: the department head, Otto Mayer, and his deputy, Michael Müller. The department member who applies for the business trip can select to whom he wants to forward his request. If he selects ‘Manager’, Otto Mayer and Michael Müller will receive the application. The first who picks it up decides on the request which in parallel is removed from the work list of the other person. As soon as an addressee is selected, the message can be sent by clicking ‘Next step’.

The subsequent state in the ‘employee’ behavior is waiting for the manager’s answer. In a receive state different messages can be expected as defined in the process model, in the example an approval or rejection. On the screen the messages which are expected and available are shown. Figure 4d depicts the screen for the subject ‘employee’ after the message ‘approved’ has arrived from the manager. As it is the only expected message in our case it can be preselected. If the user has clicked on the ‘Next step’ button the message is received and the subject proceeds to the next state.

4.3 Evaluation

The design has initially been provided in form of Microsoft PowerPoint slides and then transferred to a portlet-based user interface in order to validate it. The design prototype has been evaluated using several items. In table 1 we relate them to the psychological aspects and the design principles described by Johnson (see section 2). The list has also been influenced by [4]. Evaluating a user interface is more or less the only way to find out whether it has a chance to get accepted, which lays ground for the economic success of an entire product (see [7, p. 134]). In nearly all principles for good user interface design testing is required (see [6, p. 176] or [5]). Once testing is mentioned developers often refer to expensive usability labs. However, Jakob Nielsen showed following certain principles produces very good results with much less effort [8]. According to Nielsen five testers are sufficient for usability testing [10]. Based on this work Krug has developed a ‘lost our lease, going-out-of-business sale usability testing’ methodology as an alternative to expensive testing labs (see [7, p. 137]). Krug states ‘Testing only three users helps ensure that you will do another round soon’. Testing should be done in short intervals because testing is an iterative process.

Following those recommendations we organized test sessions with 3 users, starting with a short intro. The testers received the list of items, but we did not explicitly ask them the questions.

Table 1. Evaluation items

Psychological aspects	No.	Items
1, 2	1	Are the functions grouped according to the tasks to be executed?
1, 2	2	Is the user interface designed from normal user's perspective?
1, 2, 3	3	Are the functions described understandable?
3, 4, 5, 6	4	Are related functions grouped reasonably?
7	5	Can the user always identify his current position in the task flow?
4, 5, 6	6	Can the user always identify his next step in the task flow?
5	7	Are there needless clicks for activating important functions?
4, 5, 6	8	Are there functions supporting the repetitive execution of tasks?
3, 8	9	Does the interface allow finding required functions directly and fast?
2	10	Is there a common principle visible/sensible behind the user interface?
6	11	Is the user always informed what is going on and what needs to be done next?
5, 6	12	Does the user need to memorize many data in order to execute functions?
3, 4	13	Are the mostly used functions directly accessible?
7, 8	14	Is there a quick overview about the available functions?
1, 2, 5	15	Is there a general handling concept?
6	16	Does it take the user a long time to learn the user interface?

We explained that we just expected their recommendations on how to improve the UI according to the items list. Then we exposed the testers to the first version of the interface based on the design presented in section 4.2.

Tester 1 was a 56 years old sales person for (subject-oriented) BPM solutions. He brought in a lot of user interface experience collected from customers. The second proband was a 34 years old product manager for a BPM suite, and the third person, also 34 of age, works as a principal consultant for introducing BPM in companies.

The evaluation focused on the functional aspect (principle 2 in figure 1). Table 2 shows most significant test results and most important insights.

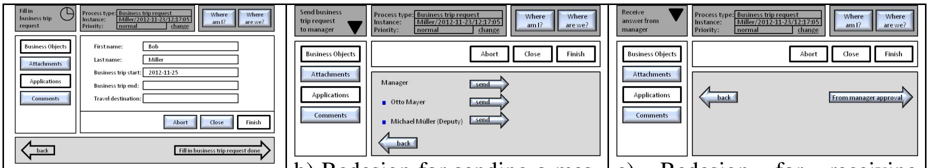
Table 2. Evaluation results

- The start screen is too complicated to understand, especially the wording (process instance, process) is confusing.
- Users want to work rather than to administer tasks or process instances
- Users want to quickly grasp what they have to do and once a task is finished they want to be informed instantly what the next steps are. This was not clear enough in the evaluated version.
- The business object to be worked on in the current task needs to be visible once the user opens a task (see figure 4b in section 4.2).
- If there is an application needed in the task its UI should be visible on the screen.
- The users need to be able to configure the interface on their own, including rearranging function groups on the screen and adapting the wording of functions as companies want to use their own labels for functions.
- Function arrangement is not consistent: Functions related to subsequent tasks should be located at same positions e.g. selecting the receiver of a message should be found at the bottom. The top line should first display information fields, followed by functions to visualize the state of a subject and of the whole process instance.

4.4 First Structural Redesign

Based on the test results we have adapted the interface structure. We show only the functional redesign due to space limits. Figure 5a depicts the modified structure for completing business object forms (function state in the behavior model). Compared to the initial design shown in figure 4b we rearranged and relabeled functions.

- **Now.** In the top line information fields on the left show the type of action (internal function), followed by information of the process instance (process type, creation date, priority). The orientation functions on the right lead to screens revealing the status of the process instance, either for the subject (where am I?) or for the entire process (where are we?). The new labels better express the meaning than 'Subject' and 'Recorder' in the previous version.
- **Needed.** In the left column functions for creating new business objects (in our example business trip request form) or adding attachments etc. can be found. The business object to be worked on is positioned in the middle part. In case a single business object is used in the process the form is opened automatically. Editing can be aborted – in this case all inputs are removed. The activity can also be interrupted – data already filled in are stored. Upon completion the form finishing is possible.
- **Next.** Activates the transition to the next state 'fill in business trip request done' in the bottom line. If the process model had specified other transitions in this state the UI would present them as additional arrows. The 'back' button leads back to the work and process list.



Process type: Business trip request
Instance: Miller/2012-11-23/12:17:05
Priority: normal

First name: Bob
Last name: Miller
Business trip start: 2012-11-25
Business trip end:
Travel destination:

Otto Mayer
Michael Müller (Deputy)

a) Redesign for a function (function state)

b) Redesign for sending a message (send state)

c) Redesign for receiving messages (receive state)

Fig. 5. Redesigned user interface

Figure 5b shows the screen for sending messages. The only activity a user needs to perform in that state is selecting the person assigned to the receiving subject in case there is more than one option. Whereas in the previous interface the user needed to check boxes and then push 'Next step' (see figure 4c), now clicking the right arrow is sufficient. If there is only one person the sending operation is executed automatically. The behavior model specifies the messages to be expected in a receive state ('Approval' or 'Rejection' in the example). At runtime it is only necessary to select one of the messages available in the current state and accept it. The arrow on the screen in figure 5c shows the approval message. If there were more messages available, the user could choose by clicking the right arrow, causing the reception. Like in the send state this procedure saves one user action (checking a box) compared to the initial interface.

5 Conclusions

Workflow engines have two UI parts: Process-independent and process-specific functions. Process-independent functions form the framework and 'infrastructure' in which the required tasks of a process are executed by users. The original UI of an existing design has been evaluated using agile user testing. Improvement potential has been identified and the UI has been changed accordingly and can be evaluated again. In future work based on the user interface approach, corresponding application programming interfaces need to be defined allowing a flexible alignment of user interfaces for workflow systems with corresponding workflow system functionality. Furthermore it has to be investigated how the agile user testing approach proposed by [8] and [7] can be integrated in a scrum-driven software development cycle, allowing for seamless interactive application development.

References

1. Cheriton, D.R.: Man-machine interface design for time-sharing systems. In: Proceedings of the ACM National Conference, pp. 362–380 (1976)
2. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., Börger, E.: Subject oriented Business Process Management. Springer, Berlin (2012)
3. Horchami, M., Fréard, D., Jamey, E., Nigay, L., Panaget, F.: A platform for designing multimodal dialogic and presentation strategies. In: Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pp. 169–170 (2007)
4. ISO 9241-110: Beurteilung von Software auf Grundlage der Internationalen Ergonomie-Norm DIN EN ISO 9241-110, `http://people.f3.htw-berlin.de/Professoren/Pruemper/instrumente/ISONORM%209241-110-L.pdf` (last access November 2012)
5. Johnson, J.: GUI bloopers 2.0: Common User Interface Design don'ts and dos. Morgan-Kaufmann Publishers, San Francisco (2007)
6. Johnson, J.: Designing with the Mind in Mind. Elsevier Publishing, Amsterdam (2010)
7. Krug, S.: Don't make me Think. New Riders Publishing, Berkley (2006)
8. Nielsen, J.: Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier, `http://www.useit.com/papers/guerrilla_hci.html` (last access November 2012)
9. Nielsen, J.: User interface directions for the web. Comm. of the ACM 42(1), 65–71 (1999)
10. Nielsen, J.: Why You Only Need to Test with 5 Users, `http://www.useit.com/alertbox/20000319.html` (last access November, 2012)
11. Nievergelt, J., Weydert, J.: Sites, modes and trails: Telling the user of an interactive system where he is, what he can do, and how to get to places. In: Guedj, R., Hagen, P., Hopgood, F., Tucker, H., Duce, D. (eds.) Methodology of Interaction, pp. 327–338. North Holland (1980)
12. Norma, D.A.: Design Rules based on Analysis of human error. Communications of the ACM 26(4), 254–258 (1983)
13. Shneiderman, B.: Designing the user Interface: Strategies for effective human-computer interaction. Addison-Wesley, Reading (1987)
14. Silver, B.: BPMN Method and Style, BPMN Implementer's Guide: A Structured Approach for Business Process Modeling and Implementation Using BPMN 2, Cody-Cassidy (2011)