
Volume 23, issue 3, May 2011
ISSN 0953-5438
ELSEVIER
BCS
Interacting
with
Computers
The Interdisciplinary Journal of
Human-Computer Interaction
Available online at www.sciencedirect.com
ScienceDirect

Interacting with Computers 23 (2011) 226–238

Contents lists available at ScienceDirect

Interacting with Computers

journal homepage: www.elsevier.com/locate/intcom

Supporting business process experts in tailoring business processes

Christian Dörner [a,1], Fahri Yetim [b,2], Volkmar Pipek [b,3], Volker Wulf [b,*]

[a] HCI Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, United States
[b] Department of Information Systems, University of Siegen, Hölderlinstraße 3, 57068 Siegen, Germany

ARTICLE INFO

Article history:
Received 27 May 2010
Received in revised form 1 March 2011
Accepted 3 March 2011
Available online 9 March 2011

Keywords:
End-user design environments
End-User Development
Empirical analysis
Composition tools
Business process modeling
Enterprise Resource Planning

ABSTRACT

Supporting end users to adapt business processes is rather uncommon in the context of large Enterprise Resource Planning systems. We present our new business process modeling environment, called SiSO, that enables business process experts to model and adapt business processes. SiSO enhances the descriptions of services that are provided by Service-Oriented Architectures. These enhanced descriptions focus on organizational-specific information, which makes it easier for business process experts to understand the capabilities of services in their organizational context. The information includes descriptions of services' functions, ratings, and keywords. SiSO's graphical user interface employs the box-and-wire UI design technique to enable business process experts to model business processes in the context of Enterprise Resource Planning systems. SiSO was qualitatively evaluated with six employees of three different companies and found useful in two application fields: (a) the visualization and automation of business processes and (b) the creation of calculations using data from different systems and sources. We think that enabling business process experts to create individual business processes is an important challenge for the design of future Enterprise Resource Planning systems.

1. Introduction

1.1. The need for tailoring business processes

Today's working environments call for flexible work practices and dynamic Enterprise Resource Planning (ERP[4]) systems in support of these practices because these systems are adaptable to environmental changes (Germonprez et al., 2007; Kawalek and Leonard, 1996; von Hippel, 2005). Developing systems that are flexible and "easy-to-adapt" by end users has been the focus of End-User Development (EUD), end-user programming, end-user computing, and (end-user) tailoring. EUD aims at developing methods, techniques, and tools that allow end users to create, modify, or extend software artifacts (Lieberman et al., 2006). Different types of end users have been identified in this research stream (Åsand and Mørch, 2006; Fischer et al., 2004; MacLean et al., 1990; Nardi and Miller, 1991; Rockart and Flannery, 1983). Among them, "super users" are seen as boundary spanners and are considered to be the best candidates for tailoring software (Åsand and Mørch, 2006; Mørch and Mehandjiev, 2000). In this paper we prefer to use the term *business process experts* to refer to such domain experts, who usually have some expertise and experience in programming and system adaptation, such as creating Excel sheets or adapting complex systems' options to their preferences.

Tailoring plays a significant role in ERP systems (Gurram et al., 2008) and requires the involvement of business process experts because they have detailed knowledge of the adaptation problems that should be addressed (Pfeiffer et al., 2008). Current literature presents arguments in favor of enabling business process experts to adapt business processes—a business process is understood as a sequence of activities that can be defined by users and executed by a machine (Melão and Pidd, 2000). One advantage of enabling business process experts to adapt business processes is the reduction of the time required to implement the changes (Agostini and Michelis, 2000).

Frequent adaptations in dynamic environments often lead to higher costs than the costs of the initial design, implementation, and introduction of the system. These costs can be cut if business process experts are enabled to do the adaptations themselves (Beringer, 2004). In addition, adaptation processes, lead by external experts, may lead to communication problems (Gallivan and Keil, 2003). We conclude that enabling business process experts to adapt systems shortens the adaptation process, cuts down costs, and increases the overall quality, while offering a high level of flexibility at the same time. Small and medium enterprises (SMEs) can benefit even more from enabling business process experts to

* Corresponding author. Tel.: +49 271 740 2910; fax: +49 271 740 3384.
E-mail addresses: cdoerner@cs.cmu.edu (C. Dörner), fahri.yetim@uni-siegen.de (F. Yetim), volkmar.pipek@uni-siegen.de (V. Pipek), volker.wulf@uni-siegen.de (V. Wulf).
[1] Tel.: +1 412 268 7565.
[2] Tel.: +49 271 740 3019; fax: +49 271 740 3384.
[3] Tel.: +49 271 740 4068; fax: +49 271 740 3384.
[4] ERP systems are big software packages consisting of different modules/subsystems that provide functionality for managing and controlling the core tasks of enterprises (e.g. human resources, production, sales, and finance).

0953-5438/$ - see front matter
doi:10.1016/j.intcom.2011.03.001

adapt systems because they have less personnel and financial resources than larger enterprises.

1.2. Gaps in current research

Professional business processes modeling tools and Service-Oriented Architectures (SOAs[5]) lack features for enabling business process experts to modify or develop business processes within their work environment. In fact, the orchestration of business processes requires technology that allows for flexible composition of system components. This is probably the main reason for why SOAs, being the latest trend in software flexibilization and reuse, have become the new design paradigm for ERP systems (Liu et al., 2008). An important example is the shift of SAP R/3 to the service-oriented SAP NetWeaver architecture (Heilig and Karch, 2008). SOAs support the modeling of business processes and a dynamic linking of services to the processes without programming these links. Based on SOAs, business process modeling environments have been developed by several companies, including IBM and SAP. However, some problems remain that are critical for enabling business process experts to modify business processes:

- First, SOAs primarily focus on the automation of machine-to-machine interaction and neglect human dimensions (e.g., no support of non-technical meta-data, such as usage descriptions of the services and their functions, ratings, and keywords).
- Second, composition tools are not designed for business process experts, who have less programming skills than professional developers.

Empirical studies report that even professional developers struggle with the implementation of SOA-based business processes. For example, service interfaces are technically oriented and do not provide non-functional and usage-related information (Brahe, 2007). This makes it harder to understand APIs (Beaton et al., 2008) as well as to find services that offer the required functionality (Aguilera et al., 2007; Hoyer et al., 2008). Visual process modeling tools are offered as a solution by leading software vendors, such as SAP, IBM, Software AG, Oracle, and TIBCO,[6] but they demand both domain expertise and advanced skills in computer use (Mørch and Mehandjiev, 2000). Such tools are unusable for business process experts, who have domain expertise, but limited skills in computer programming (e.g., typically no knowledge of XML or programming languages beyond Excel).

1.3. The purpose of this paper

We present a business process design environment for business process experts, called SiSO (Simple Service Orchestration), which enables them to model and adapt business processes in the context of ERP systems. SiSO is an effective design environment for business process experts because it uses the advantages of SOA technologies and overcomes the aforementioned deficits of supporting the human dimension of SOAs by applying UI design techniques used in mashup tools.

Our work addresses the following research questions:

(1) What are the requirements for a business process design environment for business process experts?
(2) How can these requirements be implemented effectively?
(3) How do business process experts perceive the usability and usefulness of this environment?

This article makes two major research contributions: First, it extends the focus of SOAs with respect to their human dimension by enhancing SOAs through non-technical meta-data, such as usage descriptions of the services and their functions, ratings, and keywords. Second, it provides a tool which employs the well-known box-and-wire UI design technique in a new way in order to enable business process experts (with little programming skills) to model business processes in the context of ERP systems.

1.4. Methodology

We employed different methods to answer our research questions (each method will be described in more detail in the relevant sections). The following paragraphs provide a brief overview of the methods used:

(1) The identification of the requirements followed a pluralist research approach, combining qualitative and quantitative research methods to generate richer and more reliable results (Mingers, 2001; Myers, 2009). We conducted a literature analysis to identify general requirements and two empirical studies to identify domain-specific requirements. This included a case study which explored the application domain and an online survey to validate the results of the case study.
(2) The subsequent design process of the prototype was based on a participatory design workshop (Muller, 1991, 1992) involving business process experts. Participants created mock-ups of the envisaged system, based on their knowledge of the application domain and the existing work practices.
(3) We used two different methods in combination to conduct the user tests: participatory observations and the thinking-aloud method. Business process experts from different companies used SiSO to solve the given evaluation task and verbalized their thoughts. In addition, they were interviewed about the usefulness of the tool at the end of the session.

This article is organized as follows. We will briefly discuss the related work (Section 2). Afterwards, we will focus on the three research questions in the subsequent sections: presenting the requirements analysis (Section 3), describing the development of the business process design environment (Section 4), and reporting on the evaluation of the designed prototype (Section 5). Section 6 discusses the merits, limitations, and implications of this work. Section 7 offers some conclusions and future research directions.

2. Related work

We will discuss those mashup tools that are related to SiSO and have inspired its design. Marmite is such a mashup tool, which enables end users to combine existing Web content and services from multiple Websites into new applications that were not envisaged by the websites' designers (Wong and Hong, 2007). Marmite uses a dataflow architecture that is similar to Unix pipes. It supports the extraction of content from websites (e.g., names, addresses, and dates), which can be processed in different ways, such as filtering values or adding metadata. Its output can be directed to different sinks, such as databases, map services, and web pages. However, Marmite cannot use other data sources than websites and cannot be used in an ERP context.

Yahoo Pipes is another mashup tool, available on the web. Its UI design technique is similar to Marmite. It offers predefined

[5] A Service-Oriented Architecture is a set of components that have a public interface description (usually in XML) and can be invoked via Internet protocols, such as HTTP.

[6] It should be noted that there are more and better business process modeling tools nowadays than back in 2006–2008, when this work was done.

building blocks, such as functions and data sources, which can be connected with each other to integrate, manipulate and visualize data. Its design follows as well the design of Unix pipes. Building blocks can be wired with each other, following a box-and-wire technique, to create the so-called pipes. Pipes can be shared with other users by publishing them on the Yahoo Pipes Website. Yahoo Pipes supports various formats for the presentation of the results, such as RSS, widgets or Google maps. The UI design of Yahoo Pipes inspired the design of SiSO, but the underlying architecture did not match the requirements of the ERP context we are dealing with.

Another system is EMAP (Enterprise Mashup Application Platform) which enables users to compose and integrate mashups on top of enterprise applications (Gurram et al., 2008). EMAP provides reusable widget components that are used as building blocks and can be connected with each other. In contrast to Marmite and Yahoo Pipes, EMAP uses events as communication mechanism between the different components. The components' behavior and presentation is customizable by using the components' metadata. Its enterprise focus makes it similar to SiSO, but we could not take advantage of the gained design experiences because EMAP was built parallel to SiSO.

Finally, iGoogle is, a web portal that allows its users to assemble or mash-up existing widgets into personalized websites or portals. The available widgets provide specific functions, such as weather information, news, or access to GMail. However, in contrast to the previously described mashup tools, iGoogle does not provide features for connecting the different widgets with each other to create small applications.

The described mashup tools inspired the user interface design of SiSO, but the systems differ from the design of service-based process modeling tools in the following way. Mashup tools mainly focus on visible, human-readable content (e.g., text, pictures) whereas service-based process modeling tools mainly connect business functionality (e.g., connecting production data with sales and order data) on a machine-to-machine level that is more abstract and less visible (Schroth and Janner, 2007).

Software companies started to release business process modeling tools around 2004. Nowadays, there is a huge variety of business process modeling tools available on the market. This market was much smaller during the design of SiSO (2006–2008). The market leading products included tools, such as TIBCO Business Studio, the SAP NetWeaver Composition Environment, and the IBM WebSphere Business Modeler. These products were full-featured service-oriented environments that included business process modeling tools. However, their design did not take the specific needs of business process experts into account. The tools did not address the "cognitive distance" between using and designing an application, failing to enable users with low technical skills to use these tools without getting a computer science education first.

3. Requirements analysis

To identify the requirements for SiSO, we have conducted an analysis of the relevant literature and two empirical studies: a case study and an online survey. The purpose of the literature analysis was to identify general requirements, while the empirical studies aimed to understand the application domain and to identify domain-specific requirements. In the following subsections, we describe each of the studies and provide a summary of the requirements.

3.1. Requirements from the literature study

To identify general requirements, we have analyzed literature in the fields of Service-Oriented Architectures (SOAs) and research on End-User Development (EUD). We identified several deficiencies of SOAs and investigated how they can be improved by considering principles or guidelines suggested within EUD.

As mentioned before, deficiencies of SOAs include problems, such as the incomprehensibility of information contained in the service interface (Beaton et al., 2008; Shi, 2006), the difficulties of understanding the capabilities of dynamic service compositions (Liu et al., 2008; Sirin et al., 2004; Stevens et al., 2006), and the technical orientation of service description (Abramowicz et al., 2006; Ma, 2005). These shortcomings call for the inclusion of extensions and explanations into the service descriptions (Hoyer et al., 2008).

We have also found several design techniques for end-user adaptable software systems, for example, concerning the usage of appropriate metaphors (Germonprez et al., 2007; Green and Petre, 1996; Repenning and Ioannidou, 2006), which suggest a strict congruency between the architectural concept of SOAs and the user interface. Other guidelines involve the design of familiar concepts (Germonprez et al., 2007), increasing the usability (Repenning and Ioannidou, 2006) and balancing the tradeoff between usability and powerful adaptation possibilities (Dittrich et al., 2006), as well as supporting user groups in exchanging their designed processes with each other (Mackay, 1990; MacLean et al., 1990; Nardi and Miller, 1990).

As a result of our literature review, we have identified several requirements for a business process modeling environment. The environment should have, for instance, an architecture that supports the (re-)design of business processes during use time. To make it usable for business process experts, the environment should provide an improved information structure for the description of web services. As end users often tailor collaboratively (Pipek and Kahler, 2006), the environment should provide mechanisms for sharing artifacts and for creating a collaborative documentation. In addition, the environment should use a programming language/adaptation mechanism that is closely connected to its users' mental model. The adaptation mechanism should use a minimalistic design for the elements of the modeling notation. This should be supported by an end-user-oriented design of the environment's user interface, which should make it difficult for users to commit syntactic errors.

3.2. Domain-specific requirements

We conducted a domain analysis to identify domain-specific requirements. This involved two complementary studies: (1) a case study that explored the application domain, and (2) an online survey that validated the results of the case study.

3.2.1. Exploration of the application domain: a case study

The purpose of the case study was to explore the application domain, that is, small and medium enterprises (SMEs), in order to understand business process experts' software-related problems and their adaptation needs. We interviewed eight participants from five SMEs. Table 1 provides an overview of the participating SMEs. The participants represented a broad variety of users, including department managers and employees from different departments, such as financial accounting, buying, order management or human resources. Three participants from SME B (head of the purchasing department,[7] assistant manager of the sales department, and head of the IT department), joined us again later for the design workshop (see Section 4.1), and two of them (Abby

[7] She plays an important role throughout the article. We will name her Abby to make it easier for the reader to identify her throughout the article.

Table 1
SMEs of the application domain.

	SME A	SME B	SME C	SME D	SME E
Sector	Software	Industry	Industry	Industry	Const. trade
Business	Financial software	Luggage, especially satchels	Automotive systems	Construction of claddings, floorings, etc.	Construction of roofs
Turnover	Approx. 70 million €	Approx. 40 million €	Approx. 20 million €	Approx. 2,5 million €	Approx. 2 million €
Employees	Approx. 500	155	120	12	26

and the head of the IT department) also participated in the evaluation of the system (see Section 5).

We used a qualitative research method and conducted our study in an exploratory way by means of open semi-structured face-to-face interviews, which allowed us to grasp the subjective views of the interviewees (Kvale, 1996). Two interviewers conducted the interviews. They took between 45 and 90 min each and were recorded with a digital voice-recorder. The interviews were supplemented by an ex-post interview with a department manager to analyze one selected problem in detail.

The analysis of the interviews was based on qualitative techniques (Miles and Huberman, 1994). The interviews were transcribed in abbreviated form, reflecting the main arguments. Afterwards, we analyzed them in a workshop by interpreting the transcripts and marking important statements. This allowed us to identify ex-post categories of answers that were used for sorting the statements into these categories. Through the analysis of the interviews we identified three main problems (Dörner et al., 2007):

First, *the functionality of software is often limited.* This can lead, for example, to a lack of interoperability between applications, such as data exchange problems between SAP and Excel. Participants had to navigate through multiple, and often inconsistently designed screens to gather the required data manually. One participant talked about her problems of getting the data for her Excel sheet:

> *"I sometimes have the problem that I have to access four or five things [different SAP screens] in order to get the things [data] I need; [...] for example, annually checking our customers credit limit. I need 'master data' and data from SD and some data from financial accounting and I don't get them by pressing a button."*[8]

Second, *there is a lack of tools that would allow business process experts to (semi-)automate the creation of reports and other calculations*. One participant described her problems of updating the production list twice a week:

> *"[...] We would like to have a production list directly out of SAP to avoid exporting it to Excel. [...] I have to create such a list twice a week. [...] Each time I need approximately one hour to create it [with Excel]. [...] A small thing would help us, but SAP doesn't support it. [...] It has to be programmed."*

Third, *external experts, such as consultants, are often the only persons who can adapt the software, making firms very dependent on them*. A participant talked about the company's dependence on external consultants:

> *"[...] We have to work together with a consultant to solve this issue in SAP. [...] It would be great if we could do it ourselves [without the consultant]."*

In addition, during the ex-post exploration interview with Abby (employed at firm B, see Table 1), we explored a particular, but in many ways representative work scenario in detail. The explored scenario involved the calculation of order quantities and required data from different systems (more details are provided in Section 4.2, as we used this scenario for the description of SiSO as well as for the evaluation of SiSO).

The results of the interviews showed that today's software does not provide enough flexibility to address all problems faced by users, without tailoring. Composition tools, such as business process modeling environments, could be used to offer a more efficient solution: a semi-automatic workflow. Such workflows perfectly match the requirements of planning processes because:

- Planning processes are recurrent tasks.
- Planning processes may involve data from different internal sources, such as different SAP modules and Excel.
- Planning processes may also involve data from external sources, such as production data and stock quantities.
- Planning processes involve decision logic.

3.2.2. Insights from an online survey

To confirm the validity of the results of the case study we conducted an online survey. We developed a questionnaire by following the suggestions of Fowler (2002). The questionnaire consisted of 41 questions (23 were closed questions and 18 open-ended questions). We invited 200 people via email. They were either employees of the five SMEs that participated in the case study or employees of some other SMEs to which we had good personal contacts. The participants were informed about the survey's purpose and received basic instructions for using the survey tool. The analysis of the participants' responses to the questionnaire was done by either using qualitative or quantitative methods, depending on the question types: closed questions were analyzed quantitatively and the open-ended questions qualitatively. After removing incomplete and invalid records from the data set, 69 replies remained (response rate 34.5%). The most relevant results of the online survey are:

56% of the participants said they were *familiar with describing or modeling business processes*. 40% had not made such experiences, and 4% 'did not know'. In terms of the *modeling technique used*, 46% of the participants mentioned that processes were described in a text document, 18% reported that processes were described in a table and 31% said that processes were described as diagrams. 5% 'did not know'. 38% of the participants used software for such process descriptions (they named Microsoft Visio, Excel, Word, and SAP), while 28% did not, and 34% of them did not answer.

73% of the participants said that sometimes *certain functionalities were not available* within their software or that it was cumbersome to perform certain tasks. They stated that they were adversely affected by a lack of functionality. 17% did not see any need for tailoring, and 10% 'did not know'. Participants named various reasons for the lack of functionalities, including that software vendors had implemented the software inadequately or that their programs did not work properly together, that their requirements changed frequently, and that their software had too many inappropriate functions instead of useful ones.

70% of the participants said that the *adaptation of software could help* them to solve problems, believing that adaptations improve efficiency by reducing the time needed to complete a task and by helping to avoid the creation of workarounds. 30% of the

[8] We have translated all statements from German to English.

participants disagreed, arguing that creating adaptations takes too much effort, the software cannot be adapted properly, adaptations are too expensive, and the problems are too specific and do not recur.

68% of the participants would *like to adapt their systems by themselves* while 18% would not and 14% 'did not know'. Respondents thought that self-created adaptations were useful because they could save costs, reduce the length of the adaptation process, and increase their flexibility. However, some respondents also identified insufficient knowledge and experience as an obstacle for making adaptations as well as expressing concern about compatibility problems resulting from poor adaptations.

To conclude, the results motivate the need for a business process modeling environment for business process experts and validate our findings of the case study: They indicate that the adaptation of software is viewed as helpful because it can save time and effort, which supports one of our main claims with empirical evidence. In addition, most participants expressed their preference for being able to model business processes by themselves, assuming that it would lead to an optimization of their work.

4. Design and development of the prototype

This section presents the design and development of our prototype SiSO. We chose the box-and-wire technique as UI representation of our prototype because it is a popular technique in the business process modeling domain. A popular example of box-and-wire is the Business Process Modeling Notation (BPMN) (White and Miers, 2008). We conducted a participatory design workshop, involving business process experts, in order to explore the appropriateness of box-and-wire for business process experts before its implementation. This step was necessary, since the requirements analysis neither focused on this issue nor provided detailed information needed for the implementation of box-and-wire in the given application domain. In the following subsections we first describe the design workshop and then present SiSO's functionality and architecture.

4.1. Design workshop and exploration of the box-and-wire technique

The design workshop was based on the PICTIVE concept (Muller, 1991, 1992). Designers created mock-ups of the envisaged system on the basis of their knowledge about the application domain and the existing work practices (see Section 3.2). The workshop was documented by video and audio recordings, pictures, notes, as well as the "workspace" that was created during the workshop.

Three users from one SME participated in the workshop (two of them had also participated in the interview study). They were Abby, plus the assistant manager of the sales department, and the head of the IT department. The first two could clearly be characterized as business process experts. The head of the IT department had a business background instead of a computer science background, and performed similar programming and customization tasks as the two other participants. Therefore, we consider him to be an advanced business process expert. All participants lacked experience with designing business processes, but were familiar with the customization[9] of the firm's SAP system.

Six researchers, (two designers, two moderators, and two ethnographers), attended and headed the workshop that took place in a seminar room at our university. The three participants sat at a round table and could use the following materials: paper of different formats, Post-its, differently colored pens, and the "workspace", which was basically was a large sheet of paper ($1\ m \times 1.4\ m$). Participants could use the workspace for the creation of the business process model. In addition to the mentioned materials, a blackboard and a smart board were available in the room.

4.1.1. Scenario construction and sample execution

The workshop lasted approximately 6 h and was divided into a morning session and an afternoon session. The morning session started with the discussion and analysis of several scenarios from the participants' work context, from which one was selected for the afternoon session. The selected scenario dealt with the "planning of order quantities" and was described by Abby, who had to deal with this planning task on a monthly base. The scenario had to be simplified to make it easier to understand for the other two participants and the researchers. It was used for the introduction to the concept and terminology of modeling business processes, which participants received before the afternoon session.

In the afternoon session, participants were asked to cooperatively model the scenario as business process. They could use the following design elements: boxes (small pieces of paper to be arranged on the "workspace") and wires (pen strokes between the boxes). The boxes could be labeled to specify an activity or their function. They had input and output ports which could also be labeled.

4.1.2. Analysis, results, and lessons learned

The collected materials, such as video and audio recordings, observers' notes, and the workspace (see Fig. 1), were qualitatively analyzed. The analysis revealed how users intuitively used the modeling elements and what syntax and semantics were natural for them. We will now provide more details about the results.

At the beginning of the design process, participants discussed for what the boxes should be used. They agreed to introduce several boxes to retrieve specific sub-sets of data from different SAP modules.[10] They annotated these boxes with the corresponding names of the SAP modules[11] (#1). If necessary, the center of the boxes was labeled with a selection criterion (#2) to get a subset of the data available in the SAP module. They also added a short explanation of the data that should flow out of a box (#3). Participants labeled the output ports of the boxes (#4) with the desired output format of the data, such as "Excel". They connected some output ports to the input port of a "calculation box" (#5). They had labeled this box's input ports with the abbreviated names of the data that should flow into it. The center of the box contained the formula that was used to do the calculation. Participants connected one of the other boxes' input ports to a pink colored Post-it labeled with "experience" (#6). They wanted to express that the planning process required personal "experience" as input at a certain point:

> *"It* [a certain column of the spreadsheet] is filled each year with new data [...] to converge the value. [...] It not only includes the sales volume but also an estimate. [...] It is possible that we might say: 'black becomes the new trend color'. [...] I would consider this to be experience [...], which also includes trend predictions and estimations by the marketing department. [...] It [experience] is a human factor which can't be automated."[12]

[9] They participated in adaptation processes of the firm's SAP system, which were headed and implemented by external consultants.

[10] The participants labeled several boxes as SAP modules (big components, such as human resources, or production management) or Excel spreadsheets, which suggests they wanted to use them either as data sources or sinks and not as 'resources' to get a specific business process activity done.

[11] We like to provide an explanation for the used abbreviations of the SAP modules: CO (Controlling), MM (Materials Management), SD (Sales and Distribution)

[12] We have translated all comments from German to English.

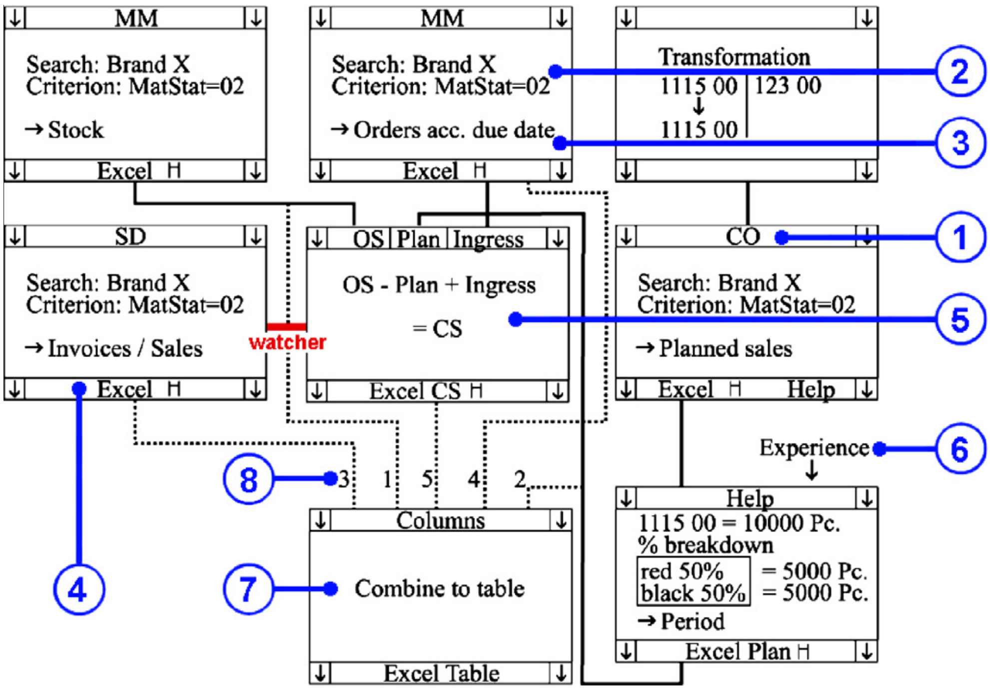

Fig. 1. Redrawn solution design created by workshop participants (reproduced from Spahn et al., 2008).

Participants connected the output ports of all relevant boxes to the input port of a box labeled "combine in table" (#7), which they wanted to use to display the result of the process as a table. They colored the wires that ended in the "combine in table" box's input ports red and annotated the wires with numbers (#8): *"Can we sort this by using red arrows? We could use them to direct it [the different output values] this way."*

Participants tried to map the business process modeling concepts to the well-known concepts of their work practice. In particular, they tried to relate everything to the Excel spreadsheet that was used in the selected scenario and even sketched the spreadsheet on a sheet of paper. The boxes in the business process served as improved data import functions for the spreadsheet. However, they also used more time- and event-related constructs. Participants created a connector, labeled "watcher" (see Fig. 1), for the automation of the created business process. The "watcher" was supposed to execute the process at the end of each month to update the data automatically:

> *"Shouldn't we define it [the overwriting of the existing values] as service? [...] It would be possible to define a date, like October 31st. [...] It [the watcher] must have an 'eye' and watch it [the time] and change the data at the right moment."*

The created business process model suggests that the box and wire technique is an appropriate composition technique for the business process modeling environment. Participants had no significant problems during the creation process. However, they used the boxes inconsistently. In some boxes, the description of functions and the description of the output were placed in the center; in others they were separated. One box did not even contain a description of the output. We think that such an inconsistent use is acceptable in the context of an early-stage design workshop.

We have identified the following design criteria for the modeling elements. The simple design of the boxes, with a separation of input, functionality, and output, seems to be adequate. Boxes were used for various purposes, e.g., as data sources, data sinks, manual inputs, and calculations, which is consistent with the elements used in business processes. The process was modeled sequentially and the decision logic was introduced in the form of different formulas within the boxes, like in the case of the "watcher", which should be activated, based on a rule.

4.2. Functionality of SiSO

SiSO's user interface is divided into three main panes, as shown in Fig. 2: search pane, modeling pane, and help pane.

4.2.1. Search pane

The search pane allows users to select the search criterion: name of service, keyword. A successful search results in a list of services. This list provides further details about services and service functions when users press the small plus buttons. Users can expand the services in this way to see which functions they provide and which parameters the functions have. There is a brief explanation (e.g., right column of the first row of the result list) for all technical information to make it more understandable.

4.2.2. Modeling pane

The modeling pane allows users to connect the functions of different services to create a process. Services are shaped as boxes having different icons depending on the data type of the port. Input-ports (yellow) and output-ports (blue) have also different colors to make it easier to distinguish between them. We implemented a simple syntax check that makes it impossible for users to connect output – with output-port or input – with input-port or to connect ports with different data types. This implementation of the design method fits with the previously introduced "tailoring by composition" method (Teege, 2000). In comparison with business process modeling tools for professionals, the modeling power of SiSO is limited, but its strength is that its cognitive demands are lower than those of professional tools.

4.2.3. Help pane

The help pane offers additional information about the services used in the two aforementioned panes. It has a toolbox that allows users to bookmark interesting services that should be used in the

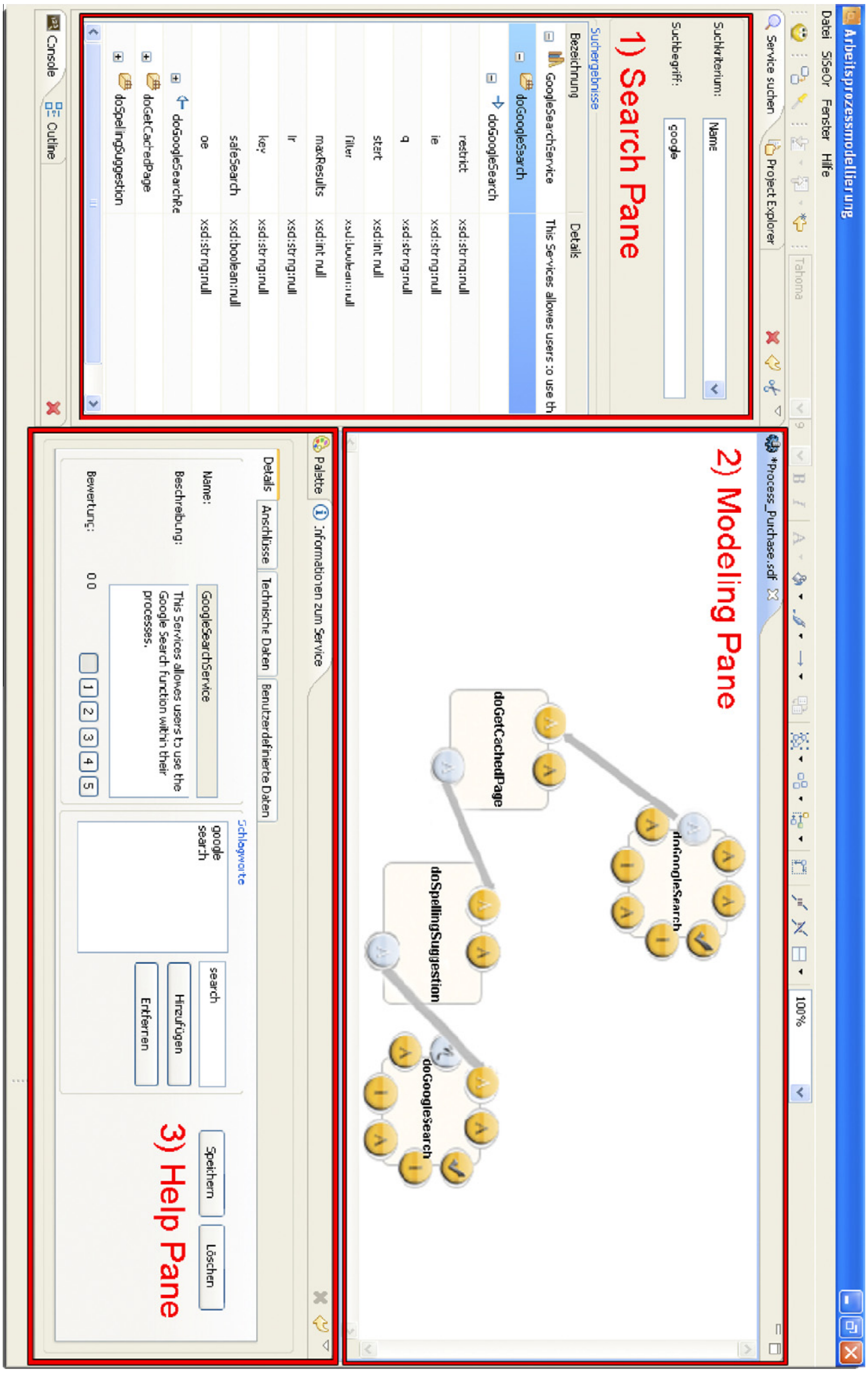

Fig. 2. SiSO's user interface.

process model, as well as to select and connect services in the modeling pane. In addition, the help pane provides explanations and information about a selected service and its ports. This information consists of usage-related information, such as usage descriptions of the services and their functions, ratings and keywords, as well as technical information such as parameter names and their data types. It is important to emphasize that the help pane allows users to edit this information.

To briefly illustrate how SiSO works, we use the scenario that we have discovered and analyzed in our exploration of the application domain (see Section 3.2.1). Abby, is head of the purchasing department and is responsible for buying all finished and semi-finished goods as well as all spare parts from her company's manufacturers in Asia. A precise planning of the order size is crucial, but is not sufficiently supported by the current SAP ERP system. Therefore, Abby has to do the planning by using an Excel sheet, which requires her to manually copy-and-paste the relevant data for her planning process from SAP to Excel.

Using SiSO can ease this cumbersome task. Abby uses SiSO's search function to find the services that provide the necessary data, including data from the company's manufacturer in Asia. Abby uses SiSO's help pane to get detailed information about some of the discovered services to find out if they provide the functionality that she is looking for. Having found the necessary services, Abby uses them for calculating the order quantities for a variety of goods. She defines a process that calculates the order quantities by connecting services on the modeling pane. The created process can be executed automatically in the future to make her planning process easier and more efficient.

4.3. Architecture of SiSO

The architecture of SiSO can be separated into three parts, as depicted in Fig. 3. The first part at the bottom is the web service layer that provides services from internal and external systems. The second part on the right side is the server component, which includes an application server and the End-User Service Orchestration Platform (EUSOP). The last part is the client component on the left side, which mainly consists of a graphical user interface. All components that were developed by our research group have a dotted background to give the reader a clear understanding of our contribution.

The web service component is based on Service-Oriented Architectures. Services can be provided by any system that is accessible via the Internet. Services must have a service interface that is defined by a WSDL (Web Service Description Language) file. WSDL describes all the pertinent information about a web service, including its name, transport binding, and message format (Ma, 2005).

The server component is built on top of the service layer. It uses open source components for the communication with the web services. Apache Tomcat is used as application server, Apache Axis2 is used as web service engine, Apache's jUDDI serves as repository, and ActiveBPEL is used as process engine.

EUSOP provides basic abstraction mechanisms from the web service technology and enhances it by providing the possibility to store additional, usage-related metadata. Its core component is the so-called generic client that offers several functions to call, use, and test web services. It interprets the elements of the WSDL files dynamically and creates stub classes for each web service. EUSOP's UDDI (Universal Description Discovery and Integration) component extends jUDDI to be able to store any kind of metadata in UDDI, which was not possible in its standard implementation. EUSOP's WSDL component serves a similar purpose as it provides enhanced documentation possibilities within the WSDL files without violating the WSDL standard.

The client component consists of two major components: EUSOP and UI. The EUSOP component makes it possible to create and test processes. The UI component is used as graphical user interface. It uses several Eclipse plug-ins as implementation basis. They provide for example the basic model of the processes and the processes' graphical representation, which consists of several web services.

Table 2 presents the requirements (determined in Section 3 and updated in Section 4.1) and how they have been implemented in SiSO.

5. Evaluation

The purpose of the evaluation study was to assess users' current views about the usability and usefulness of SiSO. Usability

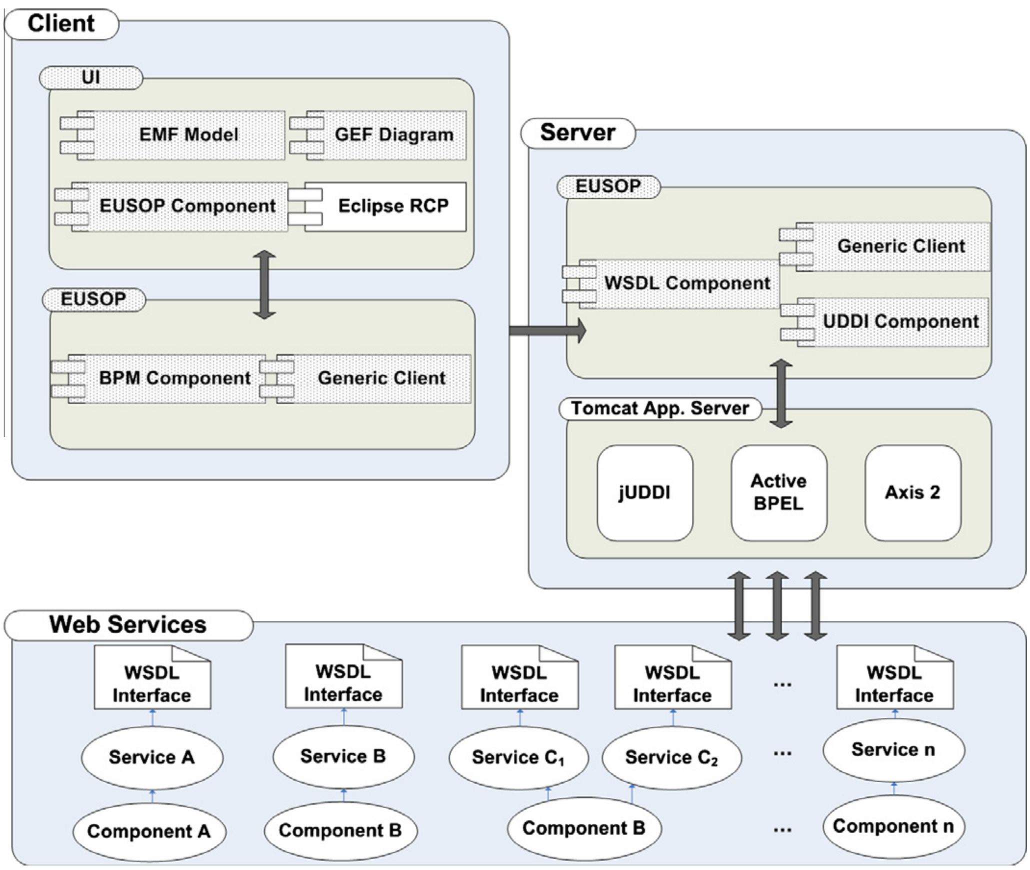

Fig. 3. SiSO system architecture.

Table 2
Implementation of the requirements.

#	Requirement	Component	Implementation
1	The system must provide an improved information structure in the WSDL files and the UDDI for the description of web services. The descriptions should be presented in a way that hides system peculiarities	Server	Enhanced the data-structures of WSDL and UDDI by adding additional fields that can be modified by the service consumers
2	The system must support the collaboration of users, e.g., by providing mechanisms for sharing artifacts and documentation	Server and client	WSDL and UDDI data is centrally managed on the server. Users can change the data by using the help pane
3	The system must have an architecture that supports the (re-)design of business processes during use time	All	Business processes are managed by the Tomcat Application Server and can be manipulated by using the graphical user interface of the client
4	The system must provide a search function for web services	Server and client	The client provides a graphical search interface that allows users to run a keyword search for services that are managed by the server's UDDI
5	The system must use a programming language, i.e. an adaptation mechanism, based on the box-and-wire technique, which it is closely connected to users' mental model	Client	The client visualizes processes based on the box-and-wire technique in the modeling pane
6	The system's user interface should follow general HCI design principles, using end-user-oriented concepts and familiar interaction techniques (e.g., point and click, hyper linking). It should also make it difficult for users to commit syntactic errors	Client	The client's graphical user interface leverages end-user-oriented concepts, as it allows users to modify business processes by changing the graphical process model. The client uses simple syntax checks based on the data types of the ports of the services to verify which ports can be connected with each other
7	The system should use a minimalistic design for the elements of the modeling notation, similar to those used in the PD workshop. In particular, the elements should be "easy-to-understand" and distinguishable	Client	The choice of the design elements was based on the results of the design workshop and have been implemented through two main modeling elements: boxes (representing services), and wires (representing the connectors of the boxes)

describes the extent to which a specified set of users can achieve specified goals with effectiveness, efficiency and satisfaction in their use context (ISO 9241-11). Usefulness describes the extent to which software actually helps to solve users' problems. Both of them are considered as fundamental determinants of user acceptance (David et al., 1999; Davis, 1989; Venkatesh et al., 2003). The evaluation focuses on the following questions:

(1) How do users perceive the usability of the environment (functions, navigations, information)?
(2) How do users perceive the usefulness of the environment?

5.1. Method, participants, tasks, and environment

We used two different methods in combination to conduct the user tests: participatory observations and the thinking-aloud method. Participants used SiSO to solve the given tasks (described below) and verbalized their thoughts while they were working on the tasks. For instance, they reported on what they were doing, which information was incomprehensible, and what they liked or disliked about the system. Users' statements helped us to identify critical use situations and to provide recommendations for improvements. In addition, we interviewed the users about the usefulness of SiSO at the end of the evaluation session.

Six employees from three different SMEs (two from each), who could be characterized as business process experts, participated in the evaluation of our system. Two of them (Abby and the head of the IT department) had participated in the case study (see Section 3.2.1) and the exploration of the box-and-wire technique (see Section 4.1). The participants had different positions in their companies, such as purchasing manager, and accounting manager. Participants had the task of using SiSO for modeling the business process, shown in Fig. 4. We consider this purchasing process to be representative for the system.

The evaluation took place in a lab at our university, as it would have been very difficult to set up SiSO and the usability testing equipment at the participants' work places. The usability testing system consisted of a screen recorder and a video camera. The video camera was placed behind the participants to record their actions in front of the computer. The video streams of the screen recorder and the camera were sent to a computer that ran a usability testing software. The software synchronized the video streams and supported the annotation of the videos.

5.2. Procedure

Before we started the sessions, we sought consent from participants to record audio, video, and take notes during the sessions. Then, we explained participants the purpose of the evaluation, and gave them an introduction to the system, the scenario, and the tasks. We asked participants to *think aloud* to verbalize their thoughts while they were working on the task (Ericsson and Simon, 1993; Robson, 2002). The first author observed the participants and asked open-ended questions to encourage responses, for instance, he asked participants what information they found incomprehensible, and what they liked and disliked. A second researcher participated, observed and took notes about critical use-situations while monitoring both video-streams on his computer. After participants had finished the task, we conducted brief semi-structured interviews with each participant. We used an interview guideline and addressed the usefulness of the systems beyond its technical and graphical design.

5.3. Data analysis

We analyzed the transcripts of the interviews and videos in light of the evaluation questions. The analysis was divided into three steps (Schmidt, 2003):

1. We structured the transcripts according to the questions of the interview guideline. Statements taken from the transcripts shed light on participants' reactions during the session.
2. Based on the transcripts, we formed ex-post categories for the analysis. This categorization included the identification of critical use situations, overall design issues, the knowledge required to use the system, organizational aspects of its implementation in the application domain, and recommendations for improvements.
3. We used these analytical categories to create a coding guideline. It helped us to cluster the data in terms of meaningful units to focus on specific problems.

6. Results

In the following we discuss the results related to usability aspects, followed by those related to SiSO's usefulness.

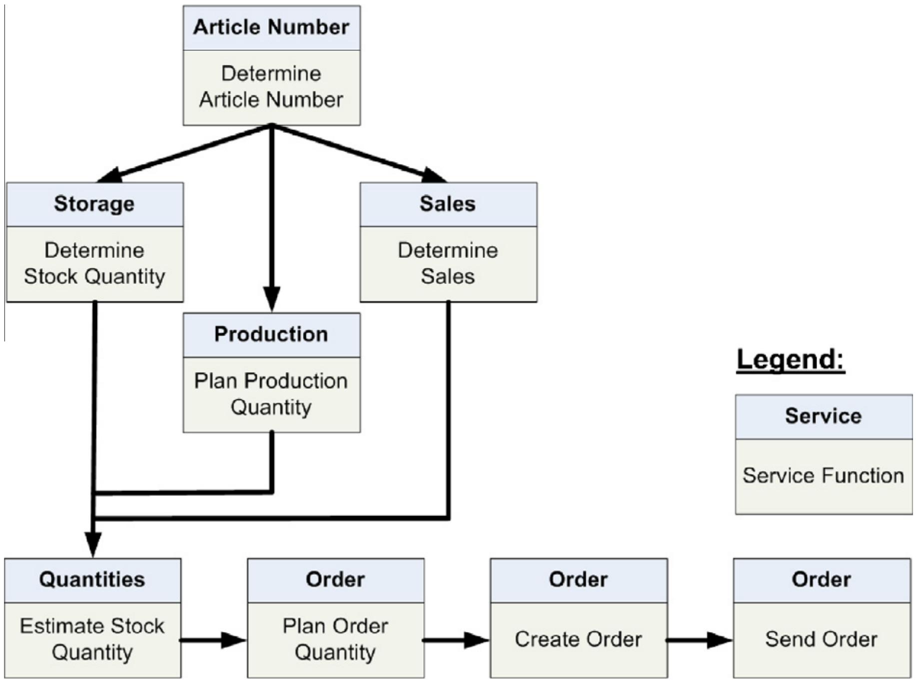

Fig. 4. Business process used for the evaluation (based on the scenario presented in Section 4.2).

6.0.1. Usability

In terms of the system's usability, we identified the following concerns. Most participants said that SiSO had a good design and was clearly structured. However, they noted that it took them some time to familiarize themselves with the system: "*I think it [the user interface] is clearly arranged. [...] I would say it is not hard to understand it.*" [13]

The representation of workflows as boxes and wires in the modeling pane followed standards. One participant said: "*I assume that boxes & wires are always used in some form for the representation of processes. I think there are not many variants.*" Data representations, such as manually entered input data, were missing and participants expected to have the option to enter "real" data into specific input ports of the functions on modeling pane: "*[...] but where is it [the input data]. I don't understand that*". In addition, participants expected to have filter functions for selecting specific sub-sets of the data provided by services: "*But I would need some possibility to tell it [the system] which articles I'm interested in; theoretically, article a, b, c; but there must be a possibility to enter that.*"

We have not implemented functions for testing particular service functions and the modeled workflow with real data. Participants expected to have such functions, because they wanted to see how functions and workflows worked to gain a better understanding: *"Well, now I would like to execute them [the functions] to see what they are doing. [...] Therefore, I probably have to connect and activate them."*

Participants had problems with the visualization of ports of a service function on the modeling pane. Ports were designed as 'circles' having different icons, but it was hard for them to tell which ports belonged together. One participant noted: *"Now, I again have the problem of seeing what can be connected with what; I can't see that here."* Participants were also not able to distinguish between ports: *"It was comprehensive, like I said. The names of the services, in particular these fields [services and ports in the modeling pane] were a little irritating."* The absence of appropriate descriptions of port names outside the help pane was problematic as well. The modeling pane indicated the data types of the ports, but not the port names: *"Where can I see which one is this one [relation of a function on the modeling to the entries in the result list]?"*

Providing extended service descriptions was one objective of SiSO's help pane. Nevertheless, participants struggled with technical terms and criticized the descriptions available, which did not tell what data was needed by a function or how the data could be processed by a function: *"This [descriptions in the help pane] was rather technical. After I found a service, there were different variants [i.e. functions provided by the service] and I could not decide [between them]. The first variant [i.e. function] calculated the amount and the second variant read it."*

One participant compared the technical nature of the descriptions in SiSO's help pane with the descriptions in the SAP Business Warehouse which were not understandable to her: *"There [in the Business Warehouse] were also only technical terms, like H-456, which are not comprehensible for 'normal' users."* She suggested extending the available information with more domain specific information that is created by the users: *"The descriptions could be complemented by personal things like: 'Warning, if you select this, you have to activate it as well'."*

Some participants mentioned that the modeling was difficult without having a runtime perspective with "real data" since they had to imagine how the system would process data. Another issue was related to the visualization of the result list of the search pane. One participant said that the arrows, used for the input- and output ports, were difficult to distinguish. A third issue was the tagging of services. One participant did not consider this functionality to be useful, while another considered it to be very useful.

Participants also had some ideas for new features for the system. They suggested the implementation of a "generic formula" service which could be used to enter mathematical formulae to perform calculations within a workflow: *"I would like to have a function that allows you to enter a formula; [...] this number multiplied by this number minus this number."* Other participants wished to have a printable version of the created models to use them for instance in meetings: *"[The models...] are often printed and discussed in meetings; therefore it would be nice to have such a function."*

[13] We have translated all statements of the participants from German to English.

One participant suggested enlarging the modeling pane because he found it too small for his workflow models. Another participant suggested that service names should be editable to be able to make them easier to understand and to fit into the work context: *"If I created this service, I would have named it differently. I would have called it 'sales'."* It was also proposed to improve the mapping between port descriptions in the result list of the search function and the ports in the modeling pane: *"I would do it like this: if I click here [function or port on the modeling pane], the corresponding row with the number should be highlighted blue [in the result list]."*

6.0.2. Usefulness

In terms of the system's usefulness we discuss participants' assessment of (1) the preconditions for implementing the system in the companies and (2) the expected effectiveness of the system.

Participants identified several preconditions that must be met before the deployment of the system in their organizations. One precondition was related to the security of the system. One participant was skeptical about the confidentiality of data coming from external systems because he does not receive a printout anymore as confirmation, which he is used to. An additional concern was the integration of SiSO with other systems, such as Excel. One participant expected to have an export function for creating Excel sheets. Furthermore, he wanted to have the option to use Excel sheets as data sources and some basic Excel functions within SiSO: *"I basically need some Excel functions like pivot tables and filters in here."* The proper documentation of services was also identified as a precondition for the successful implementation of SiSO. One participant said that this documentation should be written completely by developers and not by users. He also thought that "ordinary users", unlike him, were not very interested in service details: *"I could describe where the data comes from [...] although this should not be interesting for the user; he just wants to know: 'I enter some data at the top; I get a return at the bottom'."* One participant also mentioned that "ordinary users" should not be allowed to use the system since they do not have the knowledge to use it.

Concerning the perception of the performance of the system, most participants believed that the system could be useful in their work context. One person said the system enabled her to define workflows only once, allowing automatic execution later on. This automation could increase her efficiency: *"The effort would have to be made only once. I would have to build it [the workflow] once and then I would have less work, as I would just have to 'refresh' it."* Another participant thought the system would be useful for his product calculations, for which he needs prices of raw materials delivered by various suppliers. He said the system could increase his efficiency as he could use the system to obtain prices directly from the suppliers: *"Well, in my area I could imagine using it [the system]. When I do my calculations, I often need raw material prices from suppliers; therefore it would make sense [...] to get them directly from the supplier and not from the purchasing department."*

One participant could imagine using the system for his work in the IT department for the aggregation of data, e.g., for consolidating production data and employee data to measure their performance. Another participant could imagine to use the system in his work context (product management). However, he compared the system to another modeling tool which he used to automate a simple process and concluded that SiSO is probably easier than his current tool: *"[In the other tool...] you have to have technical knowledge to be able to use the different things; it is really a developer tool which is inappropriate for end users."* Both participants mentioned that the system could be useful in all departments where processes recur like in the production.

It should be noted that two participants were unable to assess the usefulness of the system in their real work context based on the evaluation sessions. One participant said: *"[Using the system in the work context] could be possible; that fits; [...] but I only had little time to use the system. It would be interesting for me to see how they [the workflows] really look on the screen. Right now I have only seen the formula-perspective. It would have been interesting to see the real visualization [the run-time perspective of the system]."*

7. Discussion

In this section we first discuss the lessons learned from the evaluation study and their implications for the proposed prototype. We then reflect on the general implications of this work for both research and practice, and finally describe the limitations.

7.1. Lessons Learned and Implications for the prototype

The evaluation study has shown that four out of six participants of the study could successfully complete the given task. Looking at the issues they struggled with, we first learned that users must have at least a basic understanding of the underlying concepts of business modeling in order to use SiSO effectively. Even though they do not need to be experts in business process modeling, they should know what process descriptions look like, which elements are used, and which advantages and disadvantages formal process descriptions could have. It could be argued that users need support (either training or help within the system) in order to develop a "designer mindset" (Fischer and Scharff, 2000).

Second, although participants did not perceive the system to be difficult to use, their comments indicate the need for additional improvements. For example, well-known functions, like drag and drop were missing in the current prototype. The search function caused some difficulties because it was implemented as a basic keyword search, making it difficult to obtain suitable results. Participants also suggested adding functions for filtering data because they wanted to select specific data ranges as the inputs of service functions.

Third, the design of the help pane, in particular, its structure and the technical terms used, confused the participants and made it difficult for some of them to understand the descriptions of services, service functions, and ports. In the modeling pane, participants struggled to distinguish different ports by their symbols or colors. The unclear marking made it difficult for them to determine which port could be connected to which other port.

Fourth, participants requested testing functions for services and processes, which was foreseeable, but was not the focus of this work. Testing functionalities (Dittrich et al., 2006) could be realized for example by the implementation of validity checks, exploration environments (blind) or question-based debugging systems like Whyline (Myers et al., 2004). Additional suggestions include the implementation of a function for printing the workflow models and a generic formula service for performing calculations within workflows.

Finally, with respect to the usefulness of the system, participants named two different application fields: (a) the visualization and automation of business processes and (b) the creation of calculations using data from different systems and sources. However, participants also pointed out that the successful deployment of SiSO in their organizations requires that the system provides access controls, ensures the confidentiality of data, and is compatible with the existing software infrastructure including applications like SAP and Excel.

7.2. Implications for research and practice

As our tool allows the creation of individual business processes, it raises issues concerning the quality assurance of these processes.

In some aspects, process orchestration is comparable to programming, and the available formal approaches to guarantee consistency may help here. There may be dependencies between services that have not been modeled into current service descriptions, such as user interface issues or considerations of data visibility across departments. Additional research needs to cover issues of consistency management, visualization management and user-developer collaboration in the organization.

Our work has presented an exemplary solution for the integration of organization-specific vocabulary into WSDL and UDDI. Practitioners may take our work one step further by integrating it into existing standards. In addition, the alignment of the IT infrastructure to business processes results in a convergence of technological and organizational structures. Yet, achieving the convergence through the integration of our tool is only possible if organizations are willing to invest in a new IT infrastructure based on a Service-Oriented Architecture. On the other hand, the implementation of SOAs increases the overall complexity of the IT infrastructure, as it adds additional layers of code (Rettig, 2007).

Our tool relieves business process experts from routine work, but still requires some learning efforts, i.e. involving basic knowledge about business process modeling. Organizations may provide introductory tutorials to overcome the initial barriers.

Our work uses multiple data collection techniques and combines HCI data with PD data. We used these different techniques in combination to get a detailed, domain-specific picture of the requirements before we started with the implementation. The requirements analysis helped us to identify general requirements for the implementation of the system, but did neither provide detailed, domain-specific information about the appropriateness of box-and-wire for the application domain, nor information required for the domain-specific implementation of box-and-wire. The design workshop involved users of the application domain to answer these open questions.

7.3. Limitations of the research

Our work has some limitations. In terms of the technical environment, we could not use a real ERP system to provide the services. Therefore, we used a limited number of services (about 15), which provided the necessary functionality for the evaluation task. Furthermore these services were documented by ourselves and not by people from the domain, which may have caused additional problems with regard to their comprehensibility. Finally, we should note that most of our considerations here are limited to the functionalities for individual users. We expect that the prototype will demonstrate its true value in a real setting, when particularly the strength of the "social" functionality (annotations, communication on services) can be played out.

8. Conclusions and future research

In this article we presented a prototype of a business process design environment that shows how business process experts could be allowed to model and adapt business processes. SiSO represents a new kind of system, as other business process modeling tools focus on professional developers. Our research shows how Service-Oriented Architectures can be enhanced through organizational-specific data, such as usage descriptions of the services and their functions, ratings, and keywords.

To develop the modeling environment, we first conducted a requirements analysis, consisting of a literature study and empirical studies. The empirical studies not only helped to understand the domain and to identify requirements, but also confirmed the business process experts' need for a business process modeling environment. Based on these insights, we developed the modeling environment SiSO by using Service-Oriented Architectures because of the flexibility that they offer. Finally, we evaluated the environment with business process experts. The evaluation indicated an overall satisfaction of participants with its usability and the positive assessment of its usefulness in their work context.

Designing processes is a necessary and integral part of everyday work, but it will remain a side issue for most business process experts. This results in very high demands on tools' usability; not only at the user interface level, but also in terms of maintaining consistency. In the future, it would be interesting to implement improved browsing, search, and help functions for the system. In addition, it would be interesting to implement testing functionalities for business process experts, allowing them to debug their modeled processes.

Our experiences also indicate deeper problems because service architectures were designed for professional programmers by professional programmers. While there is a long tradition of separating the interface from the application logic (here: the service structure) in software engineering methods, this is not appropriate for business process experts, who usually make sense of the service structure by what they see, i.e. the graphical user interface. We need to find ways to establish a match between the graphical interface structures and underlying service structures. Embedding the metadata which business process experts need in the WSDL files may become problematic, as the integration of usage-related documentation (e.g., screen casts describing the use of a service or application in a certain company) into the metadata structures of services will increase the download times of the WSDL files. We think that the integration of End-User Development issues in service infrastructures requires new protocols that combine experiences from the groupware/social software field with SOA approaches.

In summary, we think that this work provides some interesting design experiences that could be leveraged for the design of new business process modeling tools. Designers should take the skills and needs of business process experts into account and should include new features, such as enhanced documentation facilities that support the creation of organization-specific service descriptions.

Acknowledgements

We would like to thank all colleagues of our research group, who worked with us on this project as well as all employees of the SMEs who participated in the empirical studies. In addition, we like to thank all reviewers for their comments on the article, which helped us to present our work in an understandable and structured way. We also like to thank Brad Myers for his valuable comments on the article. The first author thanks the *Alexander von Humboldt-Foundation* for granting him a Feodor Lynen Research Fellowship for Postdoctoral Researchers. Furthermore we thank the German Federal Ministry of Education and Research (BMBF, project EUDISMES, grant number 01 IS E03 C) for funding our research.

References

Abramowicz, W., Kaczmarek, M., Kowalkiewicz, M., Zyskowski, D., 2006. Architecture for service profiling. In: Proceedings of the IEEE Services Computing Workshops. IEEE Computer Society, Chicago, Illinois, USA, pp. 121–130.

Agostini, A., Michelis, G.d., 2000. A light workflow management system using simple process models. Computer Supported Cooperative Work 9, 335–363.

Aguilera, U., Abaitua, J., Diaz, J., Bujan, D., Lopez de Ipina, D., 2007. A semantic matching algorithm for discovery in UDDI. In: International Conference on Semantic Computing (ICSC 2007). IEEE Computer Society, Irvine, CA, USA, pp. 751–758.

Åsand, H.-R.H., Mørch, A.I., 2006. Super users and local developers: the organization of end-user development in an accounting company. Journal of Organizational and End User Computing 18, 1–21.

Beaton, J., Jeong, S.Y., Xie, Y., Stylos, J., Myers, B.A., 2008. Usability challenges for enterprise service-oriented architecture APIs. In: IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). Herrsching am Ammersee, Germany, pp. 193–196.

Beringer, J., 2004. Reducing expertise tension. Communications of the ACM 47, 39–40.

Brahe, S., 2007. BPM on top of SOA: experiences from the financial industry. In: Rosemann, P.D.G.A.A.M. (Ed.), BPM2007. Springer, Heidelberg, Milan, Italy, pp. 96–111.

David, J.S., Dunn, C.L., McCarthy, W.E., Poston, R.S., 1999. The research pyramid: a framework for accounting information systems research. Journal of Information Technology 13, 7–30.

Davis, F.D., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13, 319–340.

Dittrich, Y., Lindeberg, O., Lundberg, L., 2006. End-user development as adaptive maintenance. In: Lieberman, H., Paternò, F., Wulf, V. (Eds.), End-User Development. Springer, Dordrecht, pp. 308–326.

Dörner, C., Heß, J., Pipek, V., 2007. Improving information systems by end user development: a case study. In: Proceedings of the ECIS 2007. University of St. Gallen, St. Gallen, Switzerland, pp. 783–794.

Ericsson, K.A., Simon, H.A., 1993. Protocol Analysis: Verbal Reports as Data, second ed. MIT Press, Cambridge, MA.

Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N., 2004. Meta-design: a manifesto for end-user development. Communications of the ACM 47, 33–37.

Fischer, G., Scharff, E., 2000. Meta-design: design for designers. In: Dee, S. (Ed.), Proceedings of the DIS '00. ACM Press, New York City, USA, pp. 396–405.

Fowler, F.J., 2002. Survey Research Methods, third ed. Sage Publications, Thousand Oaks, CA.

Gallivan, M.J., Keil, M., 2003. The user–developer communication process: a critical case study. Information Systems Journal 13, 37–68.

Germonprez, M., Hovorka, D., Collopy, F., 2007. A theory of tailorable technology design. JAIS 8, 351–367.

Green, T.R.G., Petre, M., 1996. Usability analysis of visual programming environments: a "cognitive dimensions" framework. Journal of Visual Languages and Computing 7, 31–174.

Gurram, R., Mo, B., Gueldemeister, R., 2008. A web based mashup platform for enterprise 2.0. In: Hartmann, S. (Ed.), WISE 2008, LCNS 5176. Springer, Berlin/Heidelberg, pp. 144–151.

Heilig, L., Karch, S., 2008. SAP NetWeaver: The Official Guide. Galileo Press, Bonn, Germany.

Hoyer, V., Stanoevska-Slabeva, K., Janner, T., Schroth, C., 2008. Enterprise mashups: design principles towards the long tail of user needs. In: Proceedings of the 2008 IEEE International Conference on Services Computing (SCC 2008). IEEE Computer Society, Honolulu, Hawaii, USA, pp. 601–602.

Kawalek, P., Leonard, J., 1996. Evolutionary software development to support organizational and business process change: a case study account. Journal of Information Technology 11, 185–198.

Kvale, S., 1996. Interviews: An Introduction to Qualitative Research Interviewing. Sage Publications, Thousand Oaks.

Lieberman, H., Paternò, F., Wulf, V., 2006. End User Development. Springer, Dordrecht.

Liu, Y., Zhuang, M., Wang, Q., Wang, H., 2008. A novel approach for service capabilities representation based on statistical study on WSDL. In: Third International Conference on Internet and Web Applications and Services (ICIW '08). IEEE Computer Society, Athens, Greece, pp. 97–102.

Ma, K.J., 2005. Web services: what's real and what's not? IT Professional 7, 14–21.

Mackay, W.E., 1990. Patterns of sharing customizable software. In: Proceedings of the CSCW '90. ACM Press, Los Angeles, California, USA, pp. 209–221.

MacLean, A., Carter, K., Lövstrand, L., Moran, T., 1990. User-tailorable systems: pressing the issues with buttons. In: Proceedings of the CHI '90. ACM Press, Seattle, Washington, USA, pp. 175–182.

Melão, N., Pidd, M., 2000. A conceptual framework for understanding business processes and business process modelling. Information Systems Journal 10, 105–129.

Miles, M.B., Huberman, A.M., 1994. Qualitative Data Analysis: An Expanded Sourcebook, second ed. Sage Publications, Thousand Oaks, CA.

Mingers, J., 2001. Combining IS research methods: towards a pluralist methodology. Information Systems Research 12, 240–259.

Mørch, A., Mehandjiev, N.D., 2000. Tailoring as collaboration: the mediating role of multiple representations and application units. Computer Supported Cooperative Work 9, 75–100.

Muller, M.J., 1991. PICTIVE – an exploration in participatory design. In: Proceedings of the CHI '91. ACM, New Orleans, LA, pp. 225–231.

Muller, M.J., 1992. Retrospective on a year of participatory design using the PICTIVE technique. In: Proceedings of the CHI '92. ACM, Monterey, CA, pp. 455–462.

Myers, B.A., Pane, J.F., Ko, A., 2004. Natural programming languages and environments. Communications of the ACM 47, 47–52.

Myers, M.D., 2009. Qualitative Research in Business & Management. Sage Publications, London.

Nardi, B.A., Miller, J.R., 1990. An ethnographic study of distributed problem solving in spreadsheet development. In: Proceedings of the CSCW '90. ACM Press, Los Angeles, California, USA, pp. 197–208.

Nardi, B.A., Miller, J.R., 1991. Twinkling lights and nested loops: distributed problem solving and spreadsheet development. International Journal of Man–Machine Studies 34, 161–184.

Pfeiffer, S., Ritter, T., Treske, E., 2008. Work Based Usability Produktionsmitarbeiter gestalten ERP-Systeme 'von unten'. Eine Handreichung, ISF aktuell, München.

Pipek, V., Kahler, H., 2006. Supporting collaborative tailoring. In: Lieberman, H., Paternò, F., Wulf, V. (Eds.), End User Development. Springer Netherlands, Berlin, pp. 327–355.

Repenning, A., Ioannidou, A., 2006. What makes end-user development tick? 13 Design guidelines. In: Lieberman, H., Paternò, F., Wulf, V. (Eds.), End User Development. Springer, Dordrecht, pp. 59–98.

Rettig, C., 2007. The trouble with enterprise software. MIT Sloan Management Review 49, 21–27.

Robson, C., 2002. Real World Research: A Resource for Social Scientists and Practitioner–Researchers, second ed. Blackwell Publishers, Oxford, UK.

Rockart, J.F., Flannery, L.S., 1983. The management of end user computing. Communications of the ACM 26, 776–784.

Schmidt, C., 2003. Analyse von Leitfadeninterviews. In: Flick, U., Kardorff, E.V., Steinke, I. (Eds.), Qualitative Forschung: Ein Handbuch. Rowohlt Taschenbuch Verlag, Hamburg, pp. 447–456.

Shi, X., 2006. Sharing service semantics using SOAP-based and REST Web services. IT Professional 8, 18–24.

Schroth, C., Janner, T., 2007. Web 2.0 and SOA: converging concepts enabling the internet of services. IT Professional 9, 36–41.

Sirin, E., Parsia, B., Hendler, J., 2004. Filtering and selecting semantic web services with interactive composition techniques. IEEE Intelligent Systems 19, 42–49.

Spahn, M., Dörner, C., Wulf, V., 2008. End user development of information artefacts: a design challenge for enterprise systems. In: Golden, W., Acton, T., Conboy, K., van der Heijden, H., Tuunainen, V.K. (Eds.), 16th European Conference on Information Systems. Galway, Ireland, pp. 303–314.

Stevens, G., Quaisser, G., Klann, M., 2006. Breakin it up: an industrial case study of component-based tailorable software design. In: Lieberman, H., Paternò, F., Wulf, V. (Eds.), End User Development. Springer Netherlands, Berlin, pp. 269–294.

Teege, G., 2000. Users as composers: parts and features as a basis for tailorability in CSCW systems. Computer Supported Cooperative Work 9, 101–122.

Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D., 2003. User acceptance of information technology: toward a unified view. MIS Quarterly 27, 425–478.

von Hippel, E., 2005. Democratizing Innovation. MIT Press, Cambridge, MA.

White, S.A., Miers, D., 2008. BPMN Modeling and Reference Guide, Future Strategies Inc., Lighthouse Point, FL, USA.

Wong, J., Hong, J.I., 2007. Making mashups with marmite: towards end-user programming for the web. In: Proceedings of the CHI '07, ACM Press, San Jose, California, USA, pp. 1435–1444.