International Journal of Cooperative Information Systems
Vol. 21, No. 4 (2012) 263–296
© World Scientific Publishing Company
DOI: 10.1142/S0218843012500049

World Scientific
www.worldscientific.com

# SUPPORTING END-USER ARTICULATIONS IN EVOLVING BUSINESS PROCESSES: A CASE STUDY TO EXPLORE INTUITIVE NOTATIONS AND INTERACTION DESIGNS

JAN HESS*, CHRISTIAN REUTER**,
VOLKMAR PIPEK‡ and VOLKER WULF§

*Information Systems and New Media, University of Siegen*
*Hoelderlinstr. 3, 57076 Siegen, Germany*
**jan.hess@uni-siegen.de*
***christian.reuter@uni-siegen.de*
‡*volkmar.pipek@uni-siegen.de*
§*volker.wulf@uni-siegen.de*

Adaptations of business processes are important in work environments, specifically when process-support needs to be tailored according to changing needs. The creation, management, and adaptation of the process models require typically modeling-experts. While these actors are knowledgeable in formalizing and operationalizing processes end-users who do not necessarily possess sophisticated modeling skills know typically local practices and framing conditions best. In this paper, we present an approach to support users in articulating their needs and to involve them into the (re-)design of process specifications. We explore how end-users reflect upon and articulate about business processes. Based on results of a qualitative study, we present a new, paper-based interaction technique, which enables users with little skills to model processes. The resulting process specifications can be transferred either in paper or in digital form into traditional modeling systems for further elaboration.

*Keywords*: Knowledge work; business process modeling; process adaptations; flexible workflows; articulation support; end-user development; pen-and-paper based interactions.

## 1. Introduction

In today's enterprises, well-designed processes guide the creation of goods and services. For traditional industries that mainly produce for the mass market, an effective and efficient workflow management is crucial. The best practices emerge with time and process descriptions represent these practices on a formal level for decision, execution and analysis purposes. At the level of knowledge work, process descriptions also guide work practices, but are characterized by a less explicit and formalized representation. Requirements depend on the context, they often change, and this requires flexible adaptations of forms that have already been established. Such types of dynamic behavior can often be found in particularly small and medium sized companies (SMEs) that usually have to react to the changing market situation rather quickly. Even if those are not necessarily called process models, here employees execute work in a process that is learned, that is being practiced, and that needs to be adapted and improved in order to be able to satisfy changing demands.

The knowledge about framing conditions and requirements for the processes often lies with several domain experts, i.e. end-users who are experienced in their area of work and act on practices established over time. In our work, such domain-experts are referenced as "end-users" with good practical experience but with no advanced skills in formal modeling. Usually, end-users understand the complexity of the process (in a sense of complexity of practice) but normally cannot influence change in the process model. Typically, the modeling process is based on complex notations and tools, which require expertise from the modeling experts. "Modeling experts", are defined as users with the knowledge to operationalize and formalize the modeling, define and structure the schedule of work. Based on that expertise, modeling experts create (complex) models with (normally) complex modeling languages that end-users have to work on. In practice, gaps will evolve between the definition (the way how work should be done) and the practice (the way how work is done). In order to bridge the gap between modeling experts and end-users,[1] we are interested in new forms of cooperation that require process notations and interaction concepts that are understandable and intuitively usable by all stakeholders involved. As a precondition, the process language needs to be easy to understand and intuitive to use.

Tools for business process reengineering, workflow management[2] or business process execution in Service-Oriented Architectures (e.g. Levardy and Browning[3]), form an infrastructure of process-oriented adaptive systems that usually require expert knowledge in their model management. Adieu[4] and SISO[5] are examples that support modeling with an easier to handle interface. However, as stated by Shipman and Marshall,[6] formal representations can be difficult to understand and can also be easily misunderstood by end-users. There is a necessity to express contextual issues through informal representations[5,6] and while the limited space of a computer screen makes simultaneous collaboration difficult, pen-based systems[7,8] provide an alternative to traditional input concepts. Participants can use different pens simultaneously and large sheets of paper provide an adequate backdrop for creating representations that include formal and informal elements.[7]

In order to support collaboration and interaction, we have chosen to focus on a scenario, where the end-user and modeling experts share a common understanding by working on a reference, which is understandable and modifiable by both. As a scenario, an end-user wants to indicate that a current process needs to be adapted, e.g. because the business process pre-defined by the system requires work that can be done in a much easier way. When articulating such improvements to the modeling expert, a common language and representation will ease the cooperation and improve flexibility. By using pen and paper, the end-user can reflect on the processes, add alternatives to a standard process and add informal elements (e.g. comments). In our scenario the activities of such articulation work has been automatically digitalized and recognized by the system. The digital representation of the design sessions can be shared with other users, e.g. to add further information, and can also be used by the modeling expert as input for the design.

In two explorative studies, we have learned how employees articulate and reflect on their work practices with the help of conceptual modeling examples from their current business. Our empirical work mainly focuses on an explorative understanding of how end-users reflect on modeling and what the representation of processes designed by end-users will look like. From pre-studies of both cases, we have summarized the requirements for a more intuitive and easy-to-understand concept of how we can involve the end-user into the process design. As a first answer to the issues mentioned above, we will present an adaptive system that allows computer supported collaborative modeling with pen and paper. The resulting representation can be used as a shared reference, similar to a shared language in software development[9] that is understandable by different stakeholders. Modeling experts will get an understanding of current practices that can be supported with well-structured models and end-users can reflect on and improve the current work by expressing their needs.

Based on the design of an end-user modeling language (Sec. 3), we will address the gap between formalizing and (re-)using the output, by following an adaptive approach based on digital pen and paper technology (Sec. 4). This should support interaction between different stakeholders by supporting the recognition of formal and informal elements, by exporting into different formats and by providing a basis for (re-)using the boundary objects further. In the discussion (Sec. 5), we further map out the importance of combining the ease of expression with the ease of interaction in the adaptation of enterprise software.

## 2. Modeling Adaptive Enterprise Systems: Integrating the End-User

The notion of enterprise software as an adaptive system that needs to evolve based on the challenges a company faces, and the active role end-users need to play in this situation, has been formulated very early on (an overview can be found in Davenport[10]): Today's systems usually follow the process-oriented perspective that has been established in the early 1990s following the success of Business Process Reengineering.[11]

### 2.1. *Business process modeling*

A "business process" is a logical set of activities leading to a special business purpose,[12] in which they are represented with different notations. In the business process modeling notation (BPMN) several categories were introduced to simplify the language and to design models on different levels of complexity. Nevertheless, BPMN was developed for technical models of business processes[12] and does not include an end-user perspective. Another important approach, the architecture of integrated information systems (ARIS),[13] aims at enabling companies to model, analyze and optimize business processes. In the field of object-oriented software engineering, the unified modeling language (UML) includes different diagram types, such as business modeling, object modeling and component modeling. All these notations seem to be quite complex. All of them have been created to support professional process and IT designers with modeling processes.

Computer systems support the process of modeling at different stages. Ellis *et al.*[14] underline the difference of the workflow model and the workflow system (execution module). While the model enables analysts and administrators to define activities and to assign them to different people, the workflow system consists of the execution environment and interface as seen by the end-users. Professional modeling tools such as the WebSphere Business Modeler, the Oracle BPEL Process Manager or the SAP NetWeaver Composition Environment are embedded in a more general business process solution with interfaces to other relevant software subsystems that support the analysis or the semi-automatic execution of pre-defined commands. Such tools are known for their high complexity, their broad range of functionalities and as such they require a rather extensive commitment to learning. For users with low technical expertise, modeling tools are difficult to use, even if supported by visual representations.[15,16] The most common ones are based on the box-and-wire metaphor that presents sequences and logical connections through lines connecting decisions or domain concepts.

## 2.2. *End-user development*

Software systems are used in different environments where the context differs from case to case. As needs and demands of users may change over time, the optimal functional range can never be completely estimated during the design process. Adaptations are necessary and become especially important at “use-time”.[17,18] Concepts of end-user development (EUD) support such flexible adaptations by enabling end-users to adapt and reconfigure systems on their own.[19] Such software systems, which are e.g. based on a component-based architecture,[20] are necessary in order to empower users in easily making their own adaptations. While EUD can be understood as an on-going process in relation to the work practice and collaborative use, Pipek and Wulf[18] and Stevens *et al.*[21] refer to the concept of “Infrastructuring”: Such conceptual framing underlines the importance of a design-in-use that involves all stakeholders in designing working infrastructures over a longer period of time.

Several EUD techniques are available to support professional and less technically experienced users. As mentioned by MacLean *et al.*[22] in their *Buttons* concept, tailoring power depends on the skills that are required to be able to do the tailoring. A tailoring culture requires flexible systems that support all users in a gentle slope. The *Buttons* system also enables the sharing of individual improvements, which benefits the whole community. Such collaboration support is often stressed as being one of the main drivers of the EUD. The system developed by Kahler[23] for example, enables a structured exchange of modifications made in Word, by using public and private repositories. In order to design a gentle slope of complexity, several EUD techniques are available. Liebermann[24] introduced the *programming by example* paradigm, which enables the capturing of an often-used series of interactions and their (re-)use with different parameters as input. Concepts such as *natural programming* address users with little or no experience in using (traditional) formal programming languages. The natural language of the focused user group has been used as input for the computer systems to realize programming options.[25] Instead of using less complex commands, forms of *visual programming* can ease (re-)creation of virtual artifacts by using representatives that are oriented at the specific application domain.[26] The interface of such systems often is realized as a construction kit that makes the (re-)positioning and connection of components that represent different activities and data possible.

### 2.3. *Process modeling for the end-user*

For end-users, changes beyond the idealized version of the preferred process are only very difficult to realize or not at all. This is problematic because business processes are not always static, nor are the work steps always the same. As already stated in Van der Aalst *et al.*,[27] it is necessary to combine the very structured and process-centered workflow domain with more (unstructured) information-centered solutions. As an approach, van der Aalst introduces adaptive workflows as a system support that is able to deal with certain changes. Such changes include individual (ad-hoc)and structural (evolutionary) changes. In order to handle dynamic changes, van der Aalst *et al.*[28] presents a generic process model approach that describes a family of variants of the same workflow process.

Mendling *et al.*[29] point out that end-users receive less support in creating process models that can be easily analyzed and understood by business modeling experts. Based on previous experience, they present seven process-modeling guidelines that help experts simplify their model and therefore make them immediately usable and more comprehensible. The work of Agostini and Michelis[30] focuses on flexible process changes for process instances by end-user themselves. In order to handle exceptions and breakdowns, the MILANO workflow system allows end-users to alter the workflow. The system supports such adaptations by enabling forward and backward jumps. Even if the end-user can change the flow of the work this way, a (well-proven) model does not need to be altered continuously.

Conceptual modeling becomes important in various areas of business life. As shown by Davies *et al.*,[31] the use of ER diagrams is the most frequently used modeling technique in practice. One of the main reasons for modeling is the support of communication among stakeholders. As language and expertise may differ between the practices, the different stakeholders have to find ways to come to a shared understanding. As a solution, boundary objects can be used to support communication between members of different communities.[32] In order to interlink the end-user and designer/developer domain, participatory design oriented methods can also support the modeling process. Of special importance is the CARD technique (Collaborative Analysis of Requirements and Design) described by Tudor *et al.*[33] Here, different cards can be (re-)arranged in collaborative sessions to (re-)design activity or task flows. The cards are (semi-)structured templates that describe an activity in more detail. Cards are usually taped on a large sheet of paper to serve as medium for other stakeholders who are also interested in the results.

Muller[34] extended the CARD approach so that it can now be used for more structured and layered participatory analyses. A similar approach is the collaborative users' task analysis (CUTA) as described by Lafreniere.[35] The color-coded cards depict activities, their duration and frequency, and are put in order on a table to create the correct schedule. Further concepts and modifications include Situation Cards,[36] Inspiration Cards[37] and the Instant Card Technique.[38] All of these techniques try to build a bridge between modeling experts and end-users, by creating a conceptual model in the sense of a boundary object. However, even if such CARD-techniques have proved to be a valuable source in creating a shared understanding, and involving users in the design, there is still a gap between formalizing and (re-)using the output of such creative sessions.

### 2.4. *Visual metaphors to support process modeling by end-users*

In order to support users in modeling, the choice of abstraction level is crucial.[39] Visual metaphors can support the appropriation of systems. Such a metaphor is related to a graphical representation with meanings similar to analogies, e.g. from the real world. As described by Hsu,[40] visual metaphors stimulate the excitement and attention of the user. In order to support visual programming, Blackwell[41] recommended using implicit metaphors. One example is the use of the dataflow model, where data moves along wires.[42] Such representations strongly relate to the box-and-wire metaphor: Functionalities or modules are represented as boxes that are connected with lines, different kinds of boxes, lines and gateways can be used to represent different types of logic. Component-based software environments make use of such representations to create a new artifact by re-using exiting modules. In such an understanding, software packages are presented as components with well-defined interfaces that can be connected with each other, without considering the precise implementation.[43]

Several tools are available that support modeling and modifying services via easy-to-use interfaces. Visual programming tools, as e.g. implemented in the FreEvolve platform,[44] support the (re-)composition by choosing relevant elements and connecting them in a meaningful way. Web 2.0 based modeling tools such as Yahoo Pipes or MS Popfly enable the creation of process descriptions called mash-ups. By adding, combining or (re)adapting web-based services, functionalities that are more complex can be realized. Daniel *et al.*[45] point out that many approaches for web service orchestration help coordinate pieces of software, but hide the human aspect. Their systems allows composition of distributed UI's. Costabile *et al.*[46] present another approach that considers the needs of different user communities in the design process of interactive systems. The adaptation of complex software systems can also be increased by the use of "pragmatic adaptive user interfaces".[47] Another example is ADIEU, developed by IBM,[48] an assistant-based tool that creates service compositions and web-based interfaces. Another approach called Simple Service Orchestration (SISO) is realized as a graphical BPEL editor and supports modeling experts in creating new service compositions[49] more easily.

Modeling on the computer screen is the most common method. However, as already shown by Nardi,[39] the limited space on the screen is a problem when it comes to visual modeling. Brainstorming sessions are also often interrupted by different actions within the physical space, e.g. by discussions, looking to each other vs. looking at the screen and using informal descriptions. In the last few years, different alternatives have been developed that make modeling more intuitive. Concepts that are based on haptic interactions have become especially important,[50] systems focusing on a more natural interaction, e.g. by using pens, enable a stronger focus on the interaction rather than on formal aspects.[8] Different techniques can be used to realize sketch-based modeling tools. In previous work whiteboards, tablets or digital paper technologies were used. The SILK system[51] enables the fast creation of electronic interfaces, by recognizing interface elements drawn with a pen. "Knight"[15] supports the collaborative modeling of UML diagrams by sketching on an electronic whiteboard. MaramaSketch[52] supports the recognition of different types of diagrams.

Pen and paper-based user interfaces bridge the gap between the virtual and physical world in order to make use of both.[7] The creation of models and designs is a creative process, very often done by first making sketches on paper.[51] Sketches of models can be created quickly and without much effort.[53] Instead of modeling in a rather formalized way, sketches on paper can easily be enriched with informal elements, such as comments or images. Un-experienced users usually contribute in this way rather than in reference to a computer-based model,[54] and this collaborative work can be supported by using large sheets of paper.[7] Other advantages are the familiarity with the media, the common access from different viewpoints and options to annotate and enrich the project in a collaborative and creative manner. On the other hand, paper is a static medium that does not offer options for feedback and flexible re-creation of the content written on it.

### 2.5. *Integrating the end-user: An interactive process*

In order to empower end-users in adapting processes they are working with, our conceptual frame distinguishes between the complexity of the process, the complexity of the model and the complexity of the language. End-users with a profound understanding of the *complexity of the processes* (their current work), normally have not expertise as modeling-experts. While formal modeling requires a different kind of expertise (abstraction, using special tools, considering the complexity of related activities, etc.), we explore alternative directions to reduce the complexity for influencing processes. Easier options for end-users to reflect and express their work can be reached by reducing the *complexity of the model*. To support such user involvement, the *complexity of the language* needs to be adapted in a way that user can make sense of it. The interface to create and change the model (includes language entities and visual representations as well) needs to be easy to understand and intuitive to use. At the same time the language needs to be powerful enough to abstract with more formal elements. The resulting model does not need to be complete in a sense that it automatically can be transferred to a system for process support. But as shared reference it can be used by modeling-experts further on to integrate (sub-)models and support them by the system.

We have seen that the challenge of allowing end-users to understand, adapt and manipulate adaptive software systems has been addressed by developing simpler notations. These nevertheless still are not to easy to use, as a certain level of complexity is necessary to define the behavior of adaptive systems, and to support the interactions between actors, covering different aspects of the expertise necessary to implement and change adaptive software. Neither of these strategies has led to a complete success in integrating end-users, so we are specifically looking at the question of how users understand and perform visual modeling using the box-and-wire-metaphor (Chapter 3), and how the interaction between end-users, more experienced users and modeling experts can be further simplified (Chapter 4).

## 3. End-User Process Adaptations: Understanding and Expressing

The aim of this study is to investigate how end-users articulate and reflect on business processes they are involved in. This work presents an empirical study that took place at an airport in Germany. We will present the adaptation practices in several departments that use a disposition system. To enable end-users to reflect on their processes and accomplish adaptations more easily, a business process modeling language is needed. Such notation should also allow end-users without professional IT training to model their processes and to change them accordingly. We conducted several explorative workshops with end-users and asked them to visualize business processes from their current work. Without pre-structuring the activities, e.g. as was done in the CARDS technique,[34] we based our exploration on a plain box-and-wire metaphor. Based on an analysis of how users represented their processes, we deduced the requirements for an end-user friendly business process representation. Based on this, we present an intuitive notation, the end-user process language (EUPL), which was evaluated in the field. The aim of this language is not to enable end-users to model formal processes, but to articulate their view to the process modelers.

### 3.1. *Empirical study*

In order to understand the organization and the practice of agents, we conducted an empirical study at a leading international service provider for the aviation industry/airport business. The company is the operator and owner of a major airport in Germany. Furthermore, they offer services in different areas of airport management at other national and international airports. “Ground handling services” is a strategic business unit and is an important revenue driver of this company. The company manages the handling of both people and luggage. They use a variety of software tools to monitor, plan, and schedule the processes that make an airport work. The systems handle some services autonomously and automatically, for other services they just deliver visualizations and notifications that allow the ground staff act accordingly. Nevertheless, the systems need to constantly be adapted as the organizational environment changes, e.g. due to new safety regulations, construction at the airport, or the changing needs of the airlines.

For our study, we focused on different actors, who deal with the software applications and their adaptations on different levels. Besides explicit requirements, we wanted to explore other implicit requirements, such as informal information about business processes. To understand the application field we used several methods: we conducted participatory observations, interviews and document analysis. In a first step, we evaluated the business analysis documents of the departments (more than 500 pages), system analysis documents, and decimations of the disposition systems and the handbooks of quality management. Furthermore, to understand current needs, we read the requirements of a new disposition system. It became clear that specific abbreviations like ATA, KSS, HOT, AVI or Off-Block-Time, Walk-Out Assistant or Ramp-Agent made it difficult to understand the entire context. Furthermore, we conducted participatory observations: Aside from unstructured participatory observations of about 45 days, we conducted structured observations of the operation management of the disposition system (three days). Additionally, we conducted six semi-structured interviews[55] with administrators of different departments ranging from 27 to 92 min, with an average time of 60 min. The interviews were recorded and transcribed. Afterwards we analyzed the material with regards to the following questions: Who does the tailoring of the disposition system (role, education, IT expertise)? What are the reasons for tailoring? How is tailoring the disposition system done? What are the limits of tailoring? How relevant is the tailoring of business processes?

### 3.2. *Tailoring in practice*

In the field, it is possible to distinguish between several different roles: *drivers*, who do operative work at the airport, like driving passengers from the airplane to the terminal, *schedulers*, who are responsible for the disposition of resources, such as the drivers and buses, and *system supporters*, who are responsible for the maintenance of the master data. After analyzing our data, we found several reasons for tailoring in the field. Some are based on new customer requirements, many are based on operational requirements, while others are based on new security regulations, easy adaptations that consider the master data of the system. Actors adapt data of cars, airplanes, airlines, airplane types and other resources. These adaptations were done by system support with the master data editor.

It is possible to distinguish between the adaptation of *master data, rules* and *business processes*. These categories differ in the power of their tailoring functionality and the tailoring expertise of the actor. MacLean *et al.*'s[22] model about the tailoring of workers, local developers and programmers, coincides with this observation. Our observed tailoring of master data, rules and processes can be transferred to this model (see Fig. 1). It shows that usually a lot of effort is needed if someone on the level of master data adaptations wants to do adaptations on the level of rules; their own staff usually does both kinds of tailoring. Adaptations on the level of processes are usually done with the help of a software development company. However, it also became obvious that the company would like to be able to do adaptations on the level of processes, thus making employees more flexible. An easy, user-friendly process-modeling notation is a precondition for easy process adaptation. This can be based on other descriptions, but it needs to be simple and easy to use for the target group of end-users, not just for developers. Therefore, the current practice and knowledge of those actors needs to be considered.

### 3.3. *Modeling workshop*

In order to support employees with different roles and expertise in modeling, a description needs to be easy to understand and use. The language also needs to be powerful enough to enable system operators with different experiences to adapt and modify processes (compare Fig. 1). To develop the requirements for a process modeling language, we used participatory design workshops.[56] These workshops were conducted at the work places of the system support. We asked three different actors to model their business processes. Although all of them are system operators,

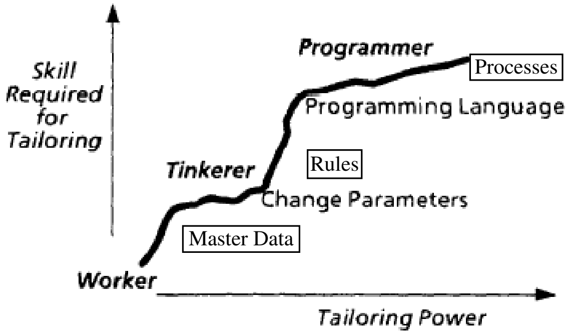


Fig. 1. Adaptation of the disposition system on the different levels of the "Taylorability Mountain" (based on MacLean *et al.* 1990[22]).

they each had different experiences in modeling. One of them had already accomplished a course on modeling, while the other two were less experienced. First, the participants reflected on often-used transport services by drawing on paper with pencils in four different colors. In the second step, the same participants had to design more complex processes with the box-and-wire metaphor, so they could use materials from the first step and then edit cards of different colors (see Fig. 2). We introduced the cards and then gave some examples. We asked the participants to draw and explain what they were doing by thinking aloud. After finishing one task, the participants were interviewed with regards to the requirements for developing a very easy-to-use process language. Therefore, it was possible to add exceptions, special cases and coordination processes to their model. The duration of each workshop was in average about 2 h.

### 3.4. *Results from modeling workshop*

The aim of the first step of the workshop (free modeling phase) was to understand how participants draw services. During the workshop, the airplane carrying process was modeled with participant A. In this process, the airplane is carried from one place to another at the airport before takeoff. Participant A was very inexperienced with process modeling and thought about it for quite a long time before he started to draw. He used boxes-and-wires: boxes for systems and then he added descriptions. Furthermore, he modeled only the standard process; he only added exceptions after he was asked to (see Fig. 3, left, red circle). Participant B modeled an inbound bus transport process. This process coordinates the transport of

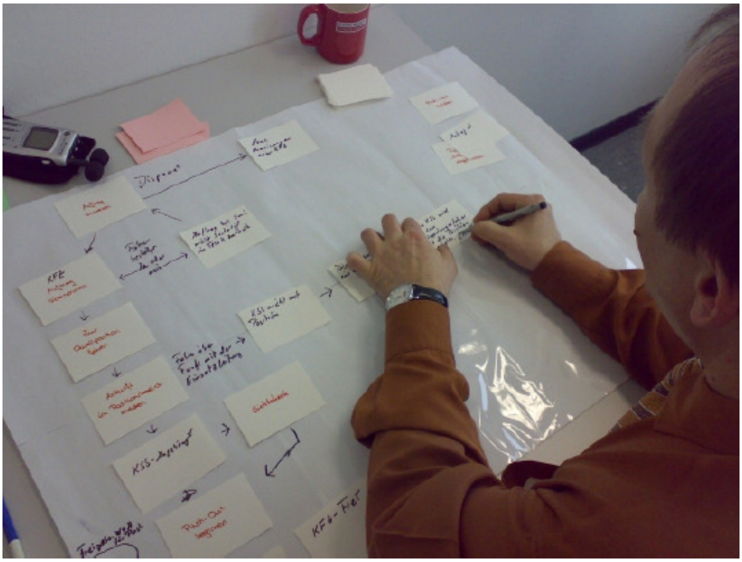


Fig. 2. Workshop with one system operator.

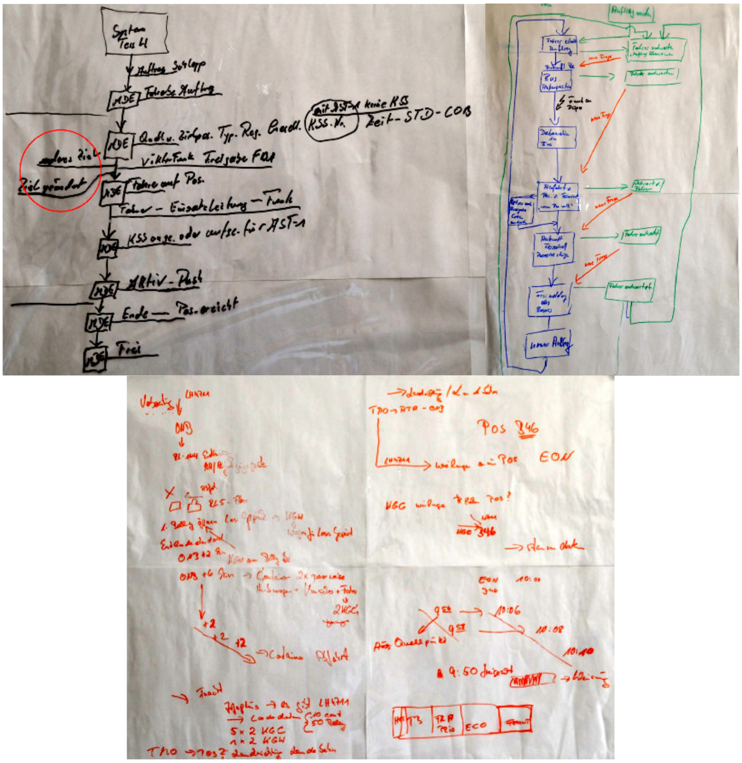


Fig. 3. Overview about the free modeling of the airplane transportation process of participant A, the inbound bus transport process of participant B, the inbound luggage transportation process of participant C.

passengers from the field to the terminal. Since he was already a bit experienced in modeling, he used boxes-and-wires in an efficient manner, and used different colors for different aspects. He first thought of how to draw the process and what he wanted to focus on, and then designed a good process model. The colors symbolized operative processes (blue), the confirmation status of the driver (green), the demanded new status (red) and the exceptions (black). Participant C modeled an inbound luggage transportation process, it contained the transport of luggage from the field to the terminal. The process was based on a storyboard and he did not use boxes and wires explicitly. He had drawn on a blackboard as done in school: The different aspects were explained in a structured way on the sheet, the result was not a process model, but a description of the process, and the different steps involved.

In the second phase of the workshops, we wanted to test the use of boxes-and-wires more explicitly. The participants were asked to use pre-defined cards to draw the processes. Participant A had to design the push-out-process, a process that describes the steps involved when an airplane has to be towed to position to be able to start. Contrary to the first task, the participant understood the aim of the second phase very quickly. To help him understand the process of modeling, some cards had already been prepared and covered existing events. After the participant was asked, he also added exceptions. The task of participant B was to design a ramp-direct-service, that brings passengers from one airplane to another, in case there is little time between flights; this process is used when a plane is late and the airport management wants to avoid passengers missing their connecting flight. The participant did not use the pre-designed cards, but designed his own cards. He used colored cards differently for different purposes. Participant C had to model a direct transfer process, which is when luggage has to be transferred from one plane to the other when passengers change planes. The participant used the cards, but did not draw connections between them. He arranged them in a line and the activities were drawn on the paper.

In general, the participants used the cards for events and activities very well. After we introduced boxes-and-wires, the quality of the results of the process modeling increased and became more structured and clear (especially for participants A and C, see Fig. 2). The box-and-wire metaphor is easier to use with pre-defined cards and should be a basis for a process modeling language. The box-and-wire metaphor seemed to be easy to handle for our participants, especially when we asked the participants to do changes in the process.

### 3.5. *Implications for the design of a common process representation*

Based on the results of the workshops, we identified several elements that needed to be supported in an end-user process language. Business processes are understood as a sequence of steps, the models contain *activities* (process steps) and *events* (their trigger or the results). Therefore, these categories need to be supported. A special event is a *gateway* to other processes. *Lines* were used to draw connections between boxes. In addition, *alternative* and *parallel processes* were also used. *Comments* contained other information relevant for the process.

None of the participants used Swimlanes, like in BPMN, which differ in the responsibilities of different actors, departments and organizations. Also, the junction and disjunction of processes like in UML, were not presented explicitly. Furthermore, XOR and OR, like in ARIS, were not used. Although many of these constructs may be necessary to maintain the logical completeness of a formal language, or to keep an overview in complex models for professional modelers, they may not be necessary to describe the needs of end-users, and sometimes that may even be confusing. Based on our findings we recommend that different aspects should be considered when developing a process language.

*R*1: *Box-and-Wire-Metaphor*: The box-and-wire metaphor is an intuitive foundation for a notation that enables system operators to reflect and model processes on their own. Even though two of the three participants used this kind of modeling without any input from us, the results were clearer in the second step when we asked the participants to use them. Also other business process languages like ARIS or BPMN use this metaphor, but with different boxes and lines. An end-user language should reduce this to the required minimum.

*R*2: *Focus on the work practice of the user*: If end-users model business processes, they mainly focus on their own work practice and model processes in which they are involved. Therefore, this process might only represents one part of the overall process and is only easy to understand for the participant herself or himself. If an overall process has to be modeled, more participants could design the process cooperatively and thus create a more objective representation. We also observed this in our workshop: in the RDS process both participants used gateways to other processes. If those processes are combined, an overall process can be displayed.

*R*3: *Events as the center of the process*: One question in the analysis of the workshop was what the center of the language could be. Based on the workshop, we think that events are the center, because those were used in every case. Between those events, activities can be added to enrich the process and to focus on specific details.

*R*4: *Slim processes*: Another very important demand of the process representation is the facility to understand the process easily. This concludes in a reduced amount of process elements to be easy to understand. Furthermore, this leads toward comments to explain difficult aspects of a process. Furthermore, a fixed direction of the process and not many exceptions make it easier to understand. If a process is too complex, a separation into several sub processes is necessary.

*R*5: *Focus on standard processes*: If end-users model processes, they should focus on the standard process. Operative business processes in the field of our study usually follow a fixed model. We found that the participants understood the process better when it followed a fixed procedure. End-users should not try to add all possible and improbable exceptions, which would make the process less easy and clear. These users are not very experienced in business process models, and therefore these exceptions would make the whole process much harder to understand for them.

*R*6: *Design from the top to the bottom*: Another question was what direction a process should follow. All processes were modeled from the top to the bottom of the paper. Process steps that followed another one, were drawn underneath the first step. It was only when the paper was full that the participants started with a new column. Alternative events were drawn horizontally.

*R*7: *Predefined events and activities*: In the second step of the workshops especially participant A, who had no experience with process modeling, used pre-defined events and activities. This enabled him to use the language in a proper way. He realized the meaning of the elements — especially recurring events, and used them. A provision of probable elements can help end-users to start modeling. This is especially important from the software adaptation perspective when existing elements have to be included into other processes.

*R*8: *Comments*: An easy process model, based on box-and-wires, cannot contain all important aspects that describe a process. Further, important aspects of the "context" can be added as comments. They can contain everything a user cannot explain with the existing elements, e.g. information about regulatory environments, examples, informal exceptions or special cases.

The derived requirements lead toward an end-user process language (EUPL). This language could empower the user to draw, change and explain their processes. The focus of this language is not on formalizing the processes in a very detailed way, but to describe the process in a language familiar to the user.

The language itself consists of four different symbols. The centers are *events*, which are drawn as blue boxes. The events are connected with *arrows*. Events can be marked with an "S", representing a *gateway*. This enables one to distinguish between different responsibilities. Other boxes represent *activities*. They are represented, as a green box with a double line on the side (see Fig. 4). They are not obligatory, but can be added between two events. It is also possible to add more than one activity between two events. This language has no specific rules, like in ARIS, where after each event an activity has to follow. In our language, events are the center and can be added with other elements. Furthermore, annotations are possible, which can combine further descriptions or required resources, actors, systems or requirements. If after one event two arrows and two events follow, this represents alternative or parallel processes. When using this language in software applications, the software should ask the user if they are parallel or alternative. We chose not to distinguish between alternative and parallel processes, because the aim was to design the language as easy as possible and not with the aim of focusing on a technical interpretation of the created processes, but to explain them to process designers, who use them to create processes in required tools.

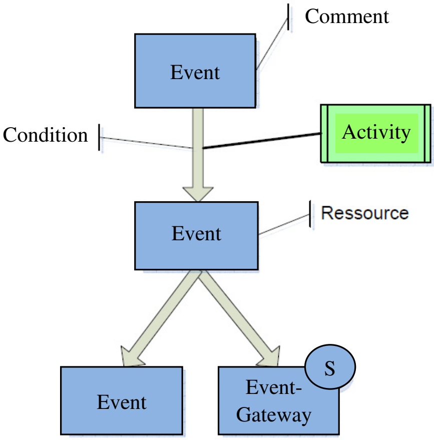


Fig. 4. Elements of EUPL.

### 3.6. *Evaluation of a common process representation*

To evaluate the end-user process language we designed the process models of the workshop participants in EUPL and gave these representations to them. Based on this, we did three interviews that lasted in average of 16 min with each participant. All participants recognized their processes and understood the symbols. One part of the process model was a brief description of the different symbols. Participant B said that the meaning of the symbols was also easy to understand without the descriptions. Participant C mentioned that different colors for activities and events were very important. Participant B said that a differentiation of elements was good, but further differentiations would be "very circumstantial" and would result in a situation where "nobody knew what is meant with a symbol". Participant A also said that more differentiation would increase circumstantial representations. All participants agreed that they could express all necessary elements related to their business processes, and that EUPL contained all important elements. Participant B proposed that all activities could have a comment showing who is doing this activity. The other two participants wanted to use this only in special cases. Participant B also proposed adding numbers to all elements, to be able to refer to them in comments. All participants agreed that events were the center of an operative business process language. All participants also recognized the alternatives in the model. Participant C said, he would just model the standard process, "which takes place in 95% of all cases" to improve the clarity. To express alternatives, participant B criticized constructs as XOR, like in ARIS, because those would always be difficult to understand.

All participants thought other co-workers would understand these processes. All of them gave positive feedback and mentioned that the symbols were easy to understand and the overall usability of the language was good.

In general, we think that a representation like this can help bring business and IT closer together by providing an easy to use language based on boxes-and-wires. Participants were able to understand the process and judged the power of the language to be big enough for their processes.

## 4. End-User Process Adaptation: Interaction Issues

As shown in the previous chapter, an easy to use representation can support end-users in the modeling of their work processes. An easy to understand notation, as presented in the form of EUPL in the chapter before, is a step toward a more intuitive process handling. As another step we focused on new interactive concepts for modeling that also support the collaborative and creative character of a group work. While modeling with paper and pen seems to be a natural way of creating a representation with formal and informal elements, we were interested in how a computer-based adaptive system can support the process. In the following chapter, we will present the results of a collaborative process modeling session with paper and pen, describe the developed adaptive system and draw implications by presenting results of a small explorative evaluation.

### 4.1. *An experiment in pen-based modeling practice*

In order to understand the practice of pen and paper based processes modeling, we conducted a pre-study with three users, who only had little experience or no experience at all in modeling. Participants should collaboratively sketch business processes on paper already known to them. In some aspects, this study has similarities with the study described in the previous chapter. However, compared to the first case where each of the participants sketched models by her- or himself, here the participants all sketched a model at the same time. Three people from a small and medium sized company (SME) with around 150 employees attended this pre-study. None of them had formal expertise in modeling business processes, but as key users of SAP they all had solid knowledge of customizing internal SAP software systems (see Table 1). Six people, involved in the research project, also attended the workshop in order to structure and guide the meeting, conduct interviews and make field-notes/images. The five-hour workshop took place at the university and was divided into two parts: an introduction and a design phase. The basis for the modeling process was a large writing pad with paper sheets the size of 1 × 1.40 m (see Fig. 5). Additionally, different kinds of office supplies were available for the modeling session, including paper sheets in several sizes, post-its, colored pencils and a board.

Table 1. Participants of the modeling workshop.

| Participant | Role | Modeling Experience |
|---|---|---|
| P1 (male) | Leader of the IT-department | SAP key user, already modeled in Microsoft Visio |
| P2 (female) | Assistant of the executive board | SAP key user, no advanced modeling expertise |
| P3 (female) | Purchasing manager | SAP key user, no advanced modeling expertise |

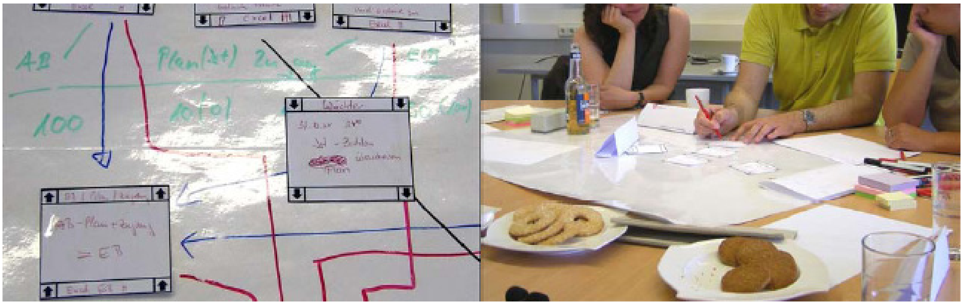


Fig. 5. Boxes and wires (left) and design workshop.

In the introduction phase, several business scenarios from the work domain of the participants, were discussed and specified. Afterwards, one of the scenarios was chosen for collaborative pen-based sketching. The chosen scenario was related to the planning of order processes. In order to get an overview of all the product groups, a complex list is necessary, which includes stock, change of stock, planned selling and current selling. The relevant data can be found in different ERP modules and needs to be combined in a structured Excel list. Within that list, additional calculations are necessary.

Before the start of the design phase, we introduced the box-and-wire model. On a large sheet of paper (1 × 1.40 meter) several actions were possible. Pre-prepared boxes cut out off paper, with an input and output port drawn on it, were used as empty entities for events and activities. It was possible to position these entities on the larger paper sheet and connect them by drawing lines between them. Additionally, annotations were possible by using pens with different colors.

At the beginning of the design phase, participants started to discuss the aim of the chosen business scenario as well as implications for modeling. As a first step, a table with columns was drawn on the paper that represented the SAP modules, where the relevant data could be found. In the following discussion, the participants started to think about how to represent the scenario on a box-and-wire level. They decided to use a separate box for each module involved. A summary of the sketched model is shown in Fig. 6. The input port of the box was annotated with keywords relating to the needed data, e.g. from SAP modules. Additionally,

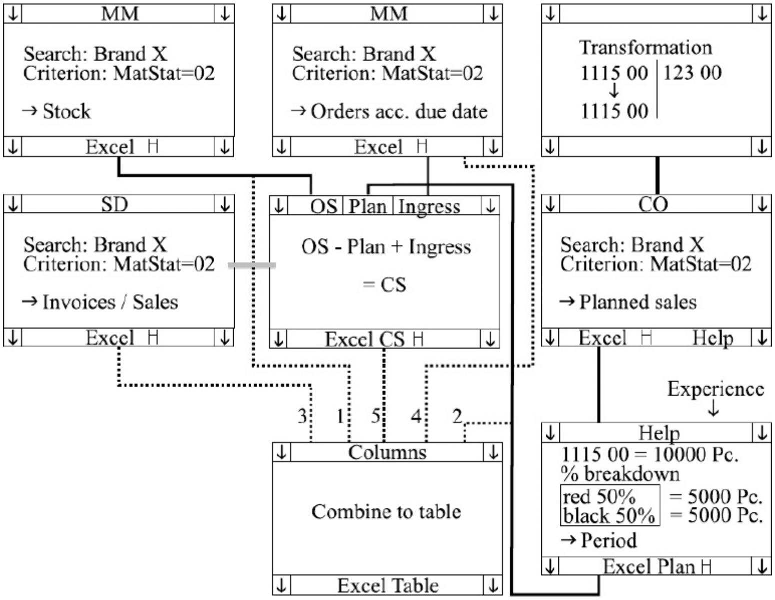


Fig. 6. Representation of the sketched model.

the participants mentioned "experience" as a further source of information. With this annotation, the participants underlined the fact that experience was necessary in order to calculate the correct planning. In this specific case, the planning process also included trend predictions and estimations from the marketing department on new trend colors. Such experience was mentioned as a human factor that cannot be automated. The output ports of the boxes were annotated with the target format (mostly Excel). Further search, selection and transformation criteria were written in the middle of the box. Participants also sketched event-driven connections between the boxes, as e.g. indicated by the grey connection line in Fig. 6. This connector was labeled as the "watcher" — a process that supports the automatic updating of data at the end of each month. Together with the output of other boxes the input for the central calculation box was directed to a final box that represented the combination of all relevant data within an Excel list. The numbers drawn on the lines going to the final box, represent the ordering of the resulting columns within the list.

As shown in the previous chapter, the workshop provided insights into how end-users without special knowledge in modeling, represent and reflect relevant business scenarios in an explorative way. By using a pen- and paper-based environment, the participants were able to "model" in an informal, collaborative and creative manner. Beside the formal elements provided (empty paper boxes), annotations (e.g. for the ordering of the columns) and informal extensions (e.g. by using "experience" as input) of the process model were used to create the process representation. The workshop also showed that the prepared boxes were used for different things, e.g. for calculation, conditions and data sources. The method to model this way, was easy to understand, and the box-and-wire metaphor supported the process quite well.

### 4.2. *Articulation support for conceptual modeling*

From the experience gathered in pre-studies, several lessons were learned. It has clearly been shown that paper-based modeling can be used as an easy-to-use and intuitive method, to let end-users express activities they are involved in. End-users — also with no or little experience in modeling — can articulate their work (and process steps) quite easily by sketching on paper and using the box-and-wire metaphor. Many advantages of such a form of modeling became obvious, so questions arose of how these paper-based modeling activities can be integrated in an IT-supported socio-technical modeling practice, where different stakeholders are involved. The focus system should support users in the collaborative creation of a model, in the transformation into a digital representation, with the extensibility and convertibility of the model, the computer generated feedback and guidance. The sketched model (the paper-based one as well as the digital representation) can be used as a boundary object that mediates between members of different Communities of Practices (CoP) — especially between end-users and modeling experts.

An ideal process flow includes three parts:

**Step 1:** End-users can reflect upon their work practice in collaborative modeling workshops. Processes are sketched on interactive paper that allows the automatic transformation into a digital (not necessarily formal) representation. Users from different CoP and with different roles can also attend, in order to understand and/or guide the scenarios.

**Step 2:** The digital representation can be exported into different formats, so that modeling experts can continue working with the material, e.g. transform it to an executable workflow model. Formal parts of the sketch as events and activities will be automatically transferred to the relating formal elements, so that modelers can check and correct them if necessary. In addition, informal elements will be recognized in order to provide meaning and remarks that are understandable to different stakeholders.

**Step 3:** The existing representations of the process (paper-based as well as the digital one) can be further used as a boundary object that bridges the end-user and the developer domain. The representation can be connected to the formal model in order to empower end-users, enabling them to inform other stakeholders about changes.

### 4.3. *Supporting modeling interactions with a paper-based approach*

As a first answer to dealing with pen-based input and transferring it to a digital representation, we would like to present an adaptive system that supports collaborative modeling. As a basic technology, we decided to use the digital pen technology from Anoto (www.anoto.com). Small cameras at the tip of the pen capture the input written on paper with a very fine dot pattern printed on it (the paper looks a bit greyish because of the pattern). The dot pattern makes every location (and every pen mark) on every sheet of paper recognizable. The captured data can then be batched or streamed to a computer via Bluetooth. By choosing this technology, every user can be part of a collaborative modeling session with his or her own pen. Just as with a standard pen, modeling can take place by sketching on paper. Several frameworks are available that support the creation of paper-based functionalities. In our case we chose the *Paper Toolkit*,[57] as an open source framework with high level API that already includes basic functionalities to support several pens, related events and interaction modes. To recognize gestures, we chose the *$1 Recognizer*,[58] which allows for fast recognition without training. The recognition of text was handled by a *Tablet PC Recognizer Pack*,[a] which is included in the *Microsoft Tablet PC Platform SDK*.[b] For the auditory feedback the *Microsoft Speech API*[c] was also used.

[a] http://www.microsoft.com/austria/windowsxp/tabletpc/muiprodguide.mspx
[b] http://msdn.microsoft.com/en-us/library/ms840465.aspx
[c] http://www.microsoft.com/speech/speech2007/speechdevarticle.mspx

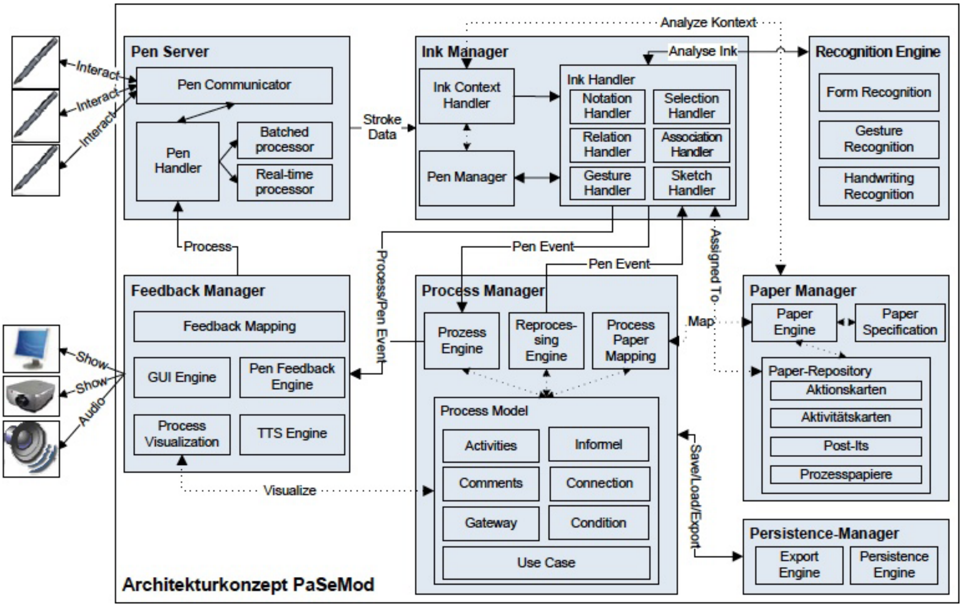


Fig. 7. Architecture of the pen and paper-based recognition system.

The architecture of the whole system is shown in Fig. 7. Input from the digital pen will be transferred to the *Pen Server*, which handles low-level communication via pen and PC and then calculates the current position. The *Pen Server* can handle and identify several pens. The stored data is then transferred to the *Ink Manager*, which is responsible for the high-level calculation of the pen input. The active context is calculated based on the type of paper (analyzed by the *Paper Manager*) used, the kind of current modeled element (analyzed by the *Recognition Engine*), the relating pen and the current pen state. The result of that calculation is directed to the *Process Manager*, which inserts the element into a digital process model. In order to connect the virtual model with the physical region on the paper, results are then remapped to the *Paper Manager*. At the end of the calculation process, the *Feedback Manager* is triggered to activate an auditory or visual feedback. The global file handling of load, save and export is done by the *Persistence-Manager*.

In order to support the creation of models in a flexible and dynamic form, the concept uses different types of paper (see Fig. 8). A "process paper" is used as background for the modeling activities (1). The size of that paper can vary and be extended beyond any screen size. Events (red) and comments (blue) are represented with post-its (4). These post-its can be positioned freely on the process paper. Drawing lines between the stickers will represent the process flow from one event to others. Events and comments can also be drawn directly on the paper by choosing the correlating process mode on the "action card" (3). However, in those cases the event cannot be replaced. Every event can be further described with a "description

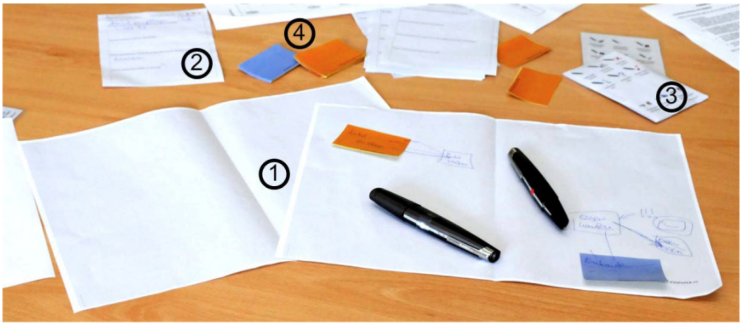


Fig. 8. Elements of digital pen and paper-based interfaces.

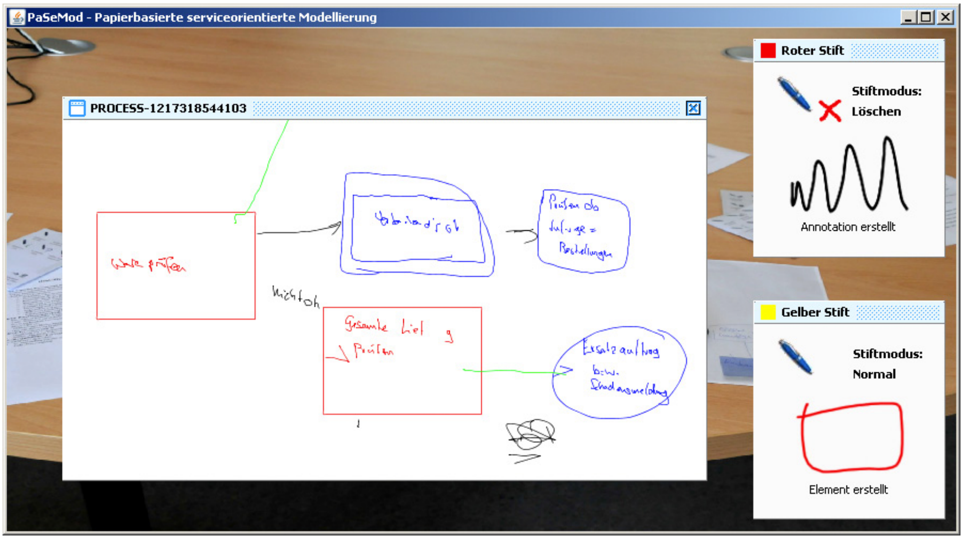
of event" (2). Such a description is realized as a pre-structured questionnaire, where further details, including name, event description, function, input and output-data, can be specified. Additionally, the already mentioned "action card" (3) provides an overview over all available process notations and pen commands. The usage of different sizes of process-paper and the simultaneous use of different pens at the same time, enable spatial arrangements in a flexible manner, e.g. in collaborative workshop settings with many end-users.

### 4.4. *Usage of the system*

One advantage of the system is the synchronous (semi-)automatic creation of a digital representation of the sketched model. In order to bridge the physical and digital domain, several design decisions had to be made for the feedback mechanism. As the digital pen does not provide sufficient feedback, a combination of visual and auditory feedback on a computer was chosen. Audio output indicates an error in the automatic recognition of the modeling process. Feedback that is more detailed is realized through feedback dialogues that are shown in the corner of the digital representation screen (as shown in Fig. 9). The pens are marked with a color that is related to the feedback dialogues on the screen. By using this color code, the current state of each pen can be displayed.

The whole model is then transferred synchronously to a digital representation as shown in Fig. 10. Events, comments and annotations are represented in different colors to ease readability and to indicate if an element was recognized correctly. Even if linked to an event, the related description of the activity will not be displayed. Every modification of the physical model needs to be explicitly defined by using different states of the pen interaction. Removing an element requires choosing the "delete" mode on the action card and to strike it out. Afterwards, the element or the link (line) can be replaced (when a post-it was used) or re-drawn.

In order to support the exchange with other end-users and with modeling experts, the digital representation of the sketched model can be exported in different formats. In the easiest form, the model can be saved as an image. Such a



PaSeMod - Papierbasierte serviceorientierte Modellierung

PROCESS-1217318544103

Roter Stift
Stiftmodus: Löschen
Annotation erstellt

Gelber Stift
Stiftmodus: Normal
Element erstellt

Ware prüfen
Prüfen der Anfrage = Bestellungen
nicht ok
Gesamte Lieferung Prüfen
Ersatzauftrag bzw. Schadensmeldung

Fig. 9. GUI with visualization of recognized elements and feedback boxes.

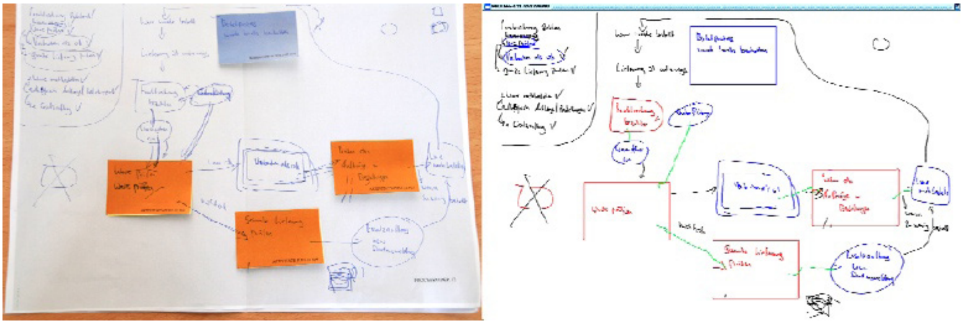


Fig. 10. Same representation in the physical (left) and digital world after recognition (right).

form of representation is easy to access by others who are interested in the domain. The model can also be used as a foundation that can be further adapted to include more complex options and integrate them into an existing IT support. The representation can also be converted into an XML file with a defined document type definition (DTD). As another option, the model can be exported to Marama in order to enable a clear separation of formal and informal elements. By using the Marama sketching toolkit,[59] the sketch can be transformed into different parts of formal elements. This way the model can be optimized and be used as an input for modeling experts who work with professional modeling tools (also compare Sec. 4.2, step 2).

The introduced system supports end-user and modeling experts in different ways. As already described in the introduction, a user may indicate changes to a given business process, e.g. as a result of an improved practice established over time. End-users mainly involved in those (sub-)tasks can meet with experts in a collaborative brainstorming session. The paper-based sketch is automatically transferred into a digital representation with formal and informal elements. Based on the representation, modeling experts can transform user-needs into a formal structure. Later, the end-user can indicate demands and needs for changes based on the sketch, e.g. in a collaborative manner, or based on the electronic representation, e.g. to highlight context-specific requirements. Concerning the importance, quality and quantity of such requests, modeling experts and decision makers can decide to adapt the model in order to synchronize it with the (best) practice. There may be different reasons for working with a shared reference, including process documentation, end-user training or quality management.

In order to adapt systems, the paper-based, as well as the digital representation, can be modified. For collaborative modifications, paper-sheets from previous brainstorming meetings can be used to remove entities or include new ones. The input is then captured and the digital representation, with the recognized formal and informal elements, will be updated as well. Changes can also be indicated individually, by changing the digital representation directly. For further work interesting functionalities may include marker for recommendation, adaptation and suggestions. The (semi-)formal digital representation can be used as input for modeling experts to transfer end-user indicated changes to the model which is executed by the system.

### 4.5. *Evaluating paper-based modeling interaction*

The computer supported pen-based modeling and interaction concept was evaluated in an expert walk through before testing it with users. In order to track down critical incidents, several scenarios were tested. Based on these results the prototype was further improved, e.g. by using a bigger font for the digital text or by providing more stable action cards made of heavier paper. After that, the system was evaluated in a two-hour workshop with two users, who had already participated in the prestudy. Both participants (P1 and P3, see also Table 1) worked in small and medium sized enterprise (SME). One of them was the manager of the IT department and already had some knowledge in modeling processes with Microsoft Visio, while the other was manager of purchase with no advanced experience in modeling on a PC. Before the evaluation started, the system was introduced to the participants. While playing around with the system, the basic functionalities, the process elements and pen states, were explained. The usage test was done in the form of a walkthrough based on a collaborative scenario, where the participants were asked to articulate their thoughts (thinking aloud). After this test, separate semi-structured interviews took place with a focus on personal opinions, satisfaction and usability. In order to analyze the study later, the evaluation was recorded on video (user test) and audio (interviews).

For the scenario-based walkthrough, participants had to model a process from their own working context in a collaborative manner using the system. In order to reflect and model freely and *ad-hoc*, the chosen scenario was related to a process from their everyday tasks. The scenario chosen included the creation of a new article in the set, which required several sub-processes, including an analysis of sales options, the creation of the article and relevant views in SAP, planning the amount of sales, the triggering of ordering processes, and a quality check and payment. While focusing on the modeling of sub-parts of the entire process, participants were asked to express their thoughts, in order to understand issues and mental reflections. During the modeling phase, the participants were seated around a large, round table with access to the pen- and paper-based material described in the previous chapter. The pens were connected to a laptop, that also provided the audio feedback. The graphical interface of the resulting representation was projected on the opposite wall via a projector.

Participants only had few problems in modeling the process with the system introduced. The syntax and the meanings of the different process elements were easy to understand and therefore intuitive to use. As shown in Fig. 11 both modes of drawing (with and without post-its) were understood and applied. However, some problems occurred, when post-its were re-positioned. It was considered to be less intuitive when the connection from the post-it to the process-paper in the background had to be removed by choosing the correlating pen mode and then crossing the small connection line. The participants expected a mode to misalign elements,

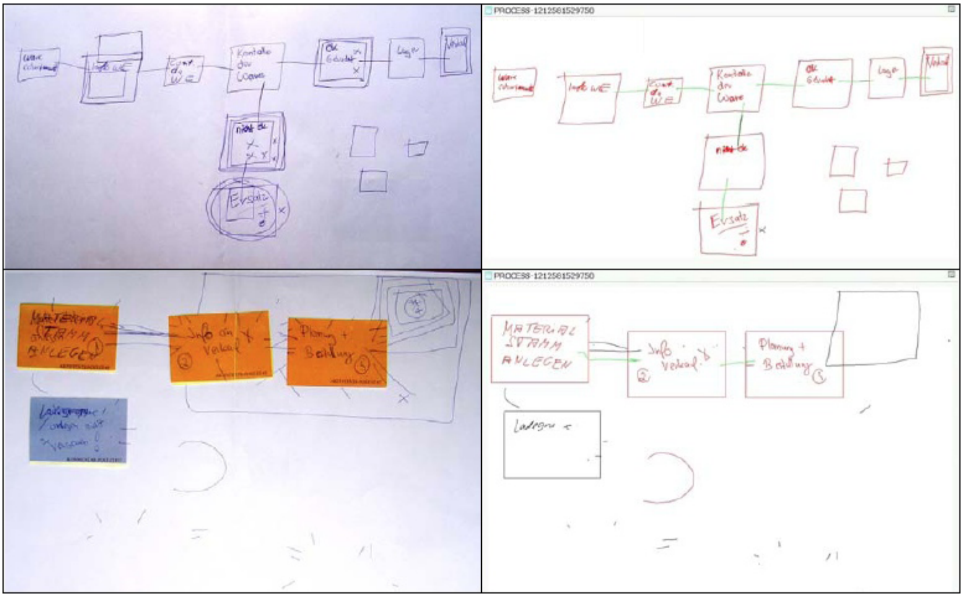


Fig. 11. Sketched models in evaluation (left) and recognized digital representation.

so that the post-it, with the correlating content, could be easily repositioned afterwards.

The good usability of the system overall, was confirmed through positive feedback from the participants. "... It is only the question of dealing with the system and practicing and then, when you know how it works, it becomes your second nature." In the beginning of the evaluation, participants had some problems trying to get up to date on the current status of the pen. Even if the status of each pen was displayed on the screen, participants did not focus on it. "For me it was clear [that there were different statuses of the pen], but I did not pay attention to them at first, even though you can see the status in the top right window". More direct feedback mechanisms, such as LEDs positioned on the pen to indicate its status, may ease the usability here. The audio feedback can support the process, if one is modeling alone; in collaborative sessions, it is more irritating, as mentioned by one participant. However, the idea of triggering a different status of the pen by choosing a symbol on the action card, and to then use the pen in different modes, was easy to understand and therefore appreciated by the participants as a "good solution" "that is easy".

As one of the main issues, the realization of the feedback has to be considered problematic. While the audio feedback was more irritating than helpful in the collaborative session, the participants focused their work on the visual feedback provided on the wall via the projector. The participants constantly checked the current status of the pen on the virtual image, which in the on-going session led to a stronger focus on the virtual representation, instead of simply brainstorming on paper. It is also important to note that participants only used formal elements (see Fig. 11) and avoided making comments as done in the pre-study. One of the users mentioned as a reason, that he tried to make a "clear" digital process model without informal elements, but that this might change depending on the task. The informal brainstorming character of the workshop was lost a bit, as the technological setting suggested a less playful modeling behavior to the participants. The results visible on the paper were also constantly checked on the wall as well. A deduction for further improvements refers to the organizational schedule of the workshop: it seems to be valuable to separate the brainstorming phase on paper (sketching) from the direction/correction phase of the recognized model (checking). Technically, our system would have also supported this type of collaboration. One of the participants could shift the elements sketched on the paper to the digital form, e.g. by typing on activities or comments with the correlating pen status.

## 5. Integrating the End-User: Combining Ease of Expression with Ease of Interaction

Systems for process modeling, as they can be found in many adaptive enterprise software tools, need to provide appropriate interactions to be able to remodel the processes for which they provide services. Software tools that support modeling are normally mainly designed for expert use. The concepts of EUD enable end-users to adapt and reconfigure systems on their own,[19] but the tailoring power depends on the skills that are required for the tailoring[22] itself and most notations have also been created to support modeling-experts.[12–14] A broad functional range that operates on complex notations enables the creation of fine-granulated formal models that guide the workflow. However, the models, normally designed by modeling experts, need to be (re-)adapted over time. Especially for knowledge work, such concepts need to be (re-)configured frequently, as processes are less structured and routinized. When a process is already modeled in a formal representation, end-users normally have no or little influence in the adaptation of that processes. Although employees may be able to articulate and describe best the process context that helps adaptive enterprise systems to deliver their services, they may not be able to model it in a formal sense. Employees can contact the modeling expert and ask for changes, but important informal aspects, e.g. that are related to the personal experience or that were identified in collaborative work practices, are not considered or can be misunderstood due to(re-)interpretation by others, and the remodeling cycles may take more time than is available. User modeling in human computer interaction[60] tries to address this issue.

The big challenge we want to overcome with our research, is how to integrate all experts necessary to adapt enterprise software (end-users as domain experts, expert modelers and mediating actors), so that their expertise can be articulated in a sustainable way. Many of the approaches we mentioned in Sec. 2 worked out well with actors on the more professional end. With our concepts and studies, we aim to improve the understanding and the options for actors at the other end of the scale. EUD techniques, such as programming by example[24] or natural programming[25] provide less complex commands. Visual programming adds virtual artifacts oriented at the application domain.[26] In our studies, we looked at the intuitiveness and appropriateness of the box-and-wire-metaphor that is considered as the basis for most visual modeling languages (Sec. 3), and we looked for different interaction modes with regard to a simpler, more collaborative modeling (Sec. 4). In two studies, employees with less experience in modeling were asked to reflect on their work practice by sketching processes and using graphical representations. As shown in both pre-studies, the pen-and paper based interaction proved to be a successful way of sketching models in a more intuitive and creative way. Participants with less technical expertise in modeling were able to reflect on their work routines in a structured manner. The box-and-wire-metaphor was confirmed as adequate, but for process modeling languages it may be helpful to provide room for structured informal model content as well, which may not be on the same level of abstraction, and may even be describing a very local use practice. Comments could be made about process executions that are just relevant for one department and not for the organization as a whole. Again, this may not help the configuration of the adaptive system directly, but it may add to a user's understanding of its inner workings or to its being embedded in the user's practice.

Supporting use and configuration of enterprise software tools by appropriation support functionality (supporting collaborative appropriation[61]) and appropriation infrastructures (means to also connect the designers[21]), can improve technology-related interactions among users as well as between users and designers. This work has demonstrated that this interaction can be the key factor in achieving both: correct configuration of software tools and also working configurations, as well as the tight-knitted integration of the changing needs of end-users into the software maintenance loop. In this contribution, we wanted to address more fundamental issues in end-user driven adaptation of process-centered enterprise infrastructures, as current tools for process modeling only provide weak support of a direct involvement of end-users.[29]

In order to integrate end-users, it is necessary to follow a dual strategy that combines "ease of expression" (using visual languages that allow end-users to express and describe the process context an adaptive system has to function in) with "ease of interaction" (providing interaction concepts that allow end-users to become creative in modeling work). An adaptive system should not only provide means to model its context or behavior, but also support the interactions that need to happen between end-users and expert modelers in order to cope with continuous changing needs in enterprise software systems. The ways of expressing processes has also become important to the point, where such representations can be (re-)used to guide others, or to provide the best practice[31] solution. Articulation support for end-users should consider formal and informal descriptions and representations for sub-areas of work practice, e.g. in supporting local groups of knowledge workers or providing the best practice examples.

Taking the findings from previous works and our pre-studies as an implication for design, we encountered the question of how a computer-based system can support such collaborative interaction by (semi-) automatically capturing and transferring it to a virtual representation as a foundation for further optimization, e.g. by directing it to experts, sharing it with others, modifying or recreating it. As a first step toward an easy solution for end-users, we presented a digital modeling tool that bridges the physical and digital domain by transferring and linking paper-based results from collaborative brainstorming sessions to a semi-structured, formalized digital output. Our prototype is able to recognize formal and informal elements written on paper and to transfer them to a digital model that can be forwarded to formal process modeling tools and be re-used later on. It is important though, to recognize that process representations do not provide the only computational basis for the services the system provides.They also become a boundary object for reflecting on activities at work, and a medium for conserving experiences. End-users can use the paper-based model as a means for mediating the communication with the modeling experts that support the adaptation in an e.g. more global framework. As an artifact, they make the complex work organization of an enterprise tangible, and the creativity that is invested during its creation forms a solid basis for the acknowledgement and the acceptance of the adaptive services. They may also foster the identification with the organization and its work practice as a whole.

The haptic interaction in the paper-based setting had particular advantages over screen-based, and even touchscreen-based, interaction. It resembled other types of creative work in enterprises that virtually all users were familiar with. The haptic qualities also added to an atmosphere of creativity that cannot be established on a ordinary computer screen, and even with large touchscreen displays the immediacy of touching and moving the material would be missing — these all are important aspects to win end-users with low familiarity of interactive technologies. The paper-based process model is also almost indefinitely extensible, as more paper can be glued and connected to the model, visibility issues due to screen size do not apply, and screen input control and turn taking is not an issue when working collaboratively with paper. The introduced system supports collaborative modeling by adapting changes synchronously in the digital visual representation. Thereby the representation looks exactly like the physical model on paper; the characteristics and creative character of the sketched output remains. By using the (re-)post-it mechanism the digital model of an already sketched process can be (re-)configured in the original physical process. Based on the tool, further interesting developments are possible. Physical paper-based elements could trigger commands for executions, e.g. a notification for modeling experts if the paper-based model was modified to articulate changes.

To better connect our approach with the work processes of expert modelers, modeling tools need to make transparent, how informal descriptions become formal. They also need to support visualizations in much simpler end-user process modeling languages that may even be ill-defined. A continuous, integrated maintenance of end-users' modeling expressions and modeling-experts' actual process models also faces new challenges. User knowledge encoded in these expressions may be less abstract and quite local to a certain usage context. While process models represent generalized "global" descriptions that will be used by enterprise systems to provide their services, the management of modeling data needs to respect these local spheres and hide comments or other informal information of local groups from each other. Adaptive systems should also show these representations if usage problems occur where these descriptions can help end-users to analyze their own mistakes, to discuss necessary adaptation or use a service breakdown to find a new requirement for the further development of the systems.

## 6. Conclusion

Modern organizations will strongly rely on IT in general, and on adaptive service infrastructures in particular. Therefore, practitioners with little or no experience in formal process modeling will need to become involved in articulation, adaptation and (re-)design activities. However, current systems are mainly designed for use by modeling experts. In order to also involve users with none ore little experience in modeling, information systems need to be designed with sufficient flexibility as well as usability to enable end-users to (re-)model process descriptions.[62] While the choice of the abstraction level is crucial,[39] visual metaphors can stimulate the excitement and attention of the user.[40] EUD approaches ease the process of service-(re)composition, e.g. FreEvolve platform[44] or Simple Service Orchestration (SISO) — a graphical BPEL editor for service orchestration,[63] but still are desigend for modeling experts. In comparison to previous work, we focus on the involvement of end-users, in the sense of domain experts with no or little experience in modeling. In order to also involve these persons in the process of flexible service adaptation, easy to understand process notations and interaction modes are necessary.

In knowledge work, processes are changing quite often, depending on the current context. Thus, easy ways to modify and comment on process descriptions are important. Our paper identified important aspects in supporting the articulation of current processes by the end-user. A case study with end users based on paper and pen indicated that besides formal aspects, informal process descriptions are of high importance. These findings underline earlier work[48,49] by highlighting the necessity to express contextual issues through informal representations. In our case we were able to show, that the use of visual languages based on the box-and-wire metaphor are helpful, but even on a language level the provision of informal and localizable structures and even free-hand drawing may contribute to making sense of the work in practice. Based on recommendations for an end-user description language, we established a concept based on a pen and paper-based interaction mode for process modeling that adds to an open, creative atmosphere and supports different modes of collaboration. The information on the physical paper-based artifacts of the brainstorming sessions, is automatically recognized and transferred into a digital representation, including formal and informal elements. Both process representations are related to each other and establish a common ground by acting as a boundary object among the different stakeholders. As an important characteristic, the system also supports (re-)use. Based on an evaluation with real practitioners from SMEs, we were able to conduct a first evaluation study, which resulted in a number of improvements, such as better separation between a sketching mode and a recognition mode, or a more appropriate audio/video feedback during modeling sessions. As a further improvement, the system may allow end-users to indicate needs for changes and improvements (on the physical and the digital model as well), which automatically triggers notifications and proposed suggestions on the side of modeling-experts.

The studies and prototypes we described have to be considered as part of a larger research effort to prepare enterprise infrastructures in supporting their own continuous development. Activities of remodeling or reconfiguration do not only affect the technological level of an infrastructure, but they also contribute to process of making sense and the appropriation of these technological artifacts. This phenomenon is conceptualized as “infrastructuring”.[18] When designing modeling languages, environments, and techniques for process specifications, these insights need to be considered to involve all levels of expertise (modeling and domain) in a better way and to include all types of experts and end-users. With our environment, we have taken a step in this direction.

## Acknowledgments

This research was supported by the German Federal Ministry for Education (BMBF) in the project EUDISMES and the German Research Foundation (DFG) in the project CONTICI. We thank Björn Borggräfe, Christian Dörner and Markus Hofmann for their contribution to this work.

## References

1. G. Fischer, Symmetry of ignorance, social creativity, and meta-design, *Knowledge-Based Syst.* **13** (2000) 527–537.
2. P. J. Kammer, G. A. Bolcer, R. N. Taylor, A. S. Hitomi and M. Bergman, Techniques for supporting dynamic and adaptive workflow, *J. Comput. Supported Coop. Work* (*CSCW*) **9** (2000) 269–292.
3. V. Levardy and T. R. Browning, An adaptive process model to support product development project management, *IEEE Trans. Eng. Management* **56** (2009) 600–620.
4. H. Shinomi and S. S. Adams, Say goodbye to complexity when developing Web services — ADIEU: The End-user Computing Tool for Web applications and Web services. *IBM ed.* (2005).
5. C. Doerner, *Tailoring Software Infrastructures: Integration of End-User Development and Service-Oriented Architecture* (Josef Eul Verlag, 2010).
6. F. M. I. Shipman and C. C. Marshall, Formality considered harmful: Experiences, emerging themes, and directions on the use of formal representations in interactive systems, *J. Comput. Supported Coop. Work* **8** (1999) 333–352.
7. G. Dai, Pen-based user interface, *Proc. 8th Int. Conf. Computer Supported Cooperative Work in Design* (2004), pp. 32–36.
8. J. Subrahmonia and T. Zimmerman, Pen computing: Challenges and applications, *Proc. 15th Int. Conf. Pattern Recog.* **62** (2000) 60–66.
9. B. Hedberg and S. Jönsson, Designing semiconfusing information systems for changing organizations, *Account. Org. Soc.* **3** (1978) 47–64.
10. T. H. Davenport, *Process Innovation: Reengineering Work Through Information Technology* (Harvard Business School Press, Boston, MA, USA, 1993).
11. C. Mielke, *Geschäftsprozesse — UML — Modellierung und Anwendungs-Generierung* (Spektrum Akademischer Verlag, Heidelberg, 2002).
12. S. A. White, *Business Process Modeling Notation* (*BPMN*) *Specification 1.0.* (2004).
13. A.-W. Scheer, *Architektur Integrierter Informations Systeme* (Springer, Berlin, 1992).
14. C. Ellis, K. Keddara and G. Rozenberg, Dynamic change within workflow systems, *Proc. Conf. Org. Comput. Syst. COCS* **95** (1995) 10–21.
15. T. Schümmer, S. Lukosch and R. Slagter, Using patterns to empower end-users — The oregon software development process for groupware, *Int. J. Cooperative Inf. Syst.* **15** (2006) 259–288.
16. X. Liu, G. Huang and H. Mei, Towards end-user service composition, *Proc. 31st Annual Int. Computer Software and Applications Conf.*, Vol. 1 (COMPSAC 2007) **1** (2007).

17. G. Fischer and E. Scharff, Meta-design — Design for designers, *Proc. Int. Conf. Designing Interactive Systems* (2000), pp. 396–405.
18. V. Pipek and V. Wulf, Infrastructuring: Toward an integrated perspective on the design and use of information technology, *J. Assoc. Inf. Syst.* (*JAIS*) **10** (2009) 447–473.
19. H. Lieberman, F. Paterno and V. Wulf, *End-User Development* (Springer, Dordrecht, The Netherlands, 2006).
20. V. Wulf, V. Pipek and M. Won, Component-based tailorability: Enabling highly flexible software applications, *Int. J. Human-Computer Studies* **66** (2008) 1–22.
21. G. Stevens, V. Pipek and V. Wulf, Appropriation infrastructure: Supporting the design of usages, *Proc. Second Int. Sympo. End-User Development* (*IS-EUD 2009*) (2009) pp. 50–69.
22. A. MacLean, K. Carter, L. Lövstrand and L. Moran, User-tailorable systems: Pressing the issues with buttons, *Proc. SIGCHI Conf. Human Factors in Computing Systems: Empowering People*, Seattle, Washington, United States (1990).
23. H. Kahler, More than WORDs — Collaborative tailoring of a word processor, *J. Univ. Comput. Sci.* (*j.ucs*) **7** (2001) 826–847.
24. H. Lieberman, *Your Wish is My Command: Programming by Example* (Morgan Kaufmann, 2001).
25. B. A. Myers, J. F. Pane and A. Ko, Natural programming languages and environments, *Commun. ACM* **47** (2004) 47–52.
26. N. Mehandjiev and L. Bottaci, User-enhanceability for organizational information systems through visual programming, *Proc. 8th Int. Conf. Advances Information System Engineering* (1996).
27. W. M. P. van der Aalst, T. Basten, H. M. W. Verbeek, P. A. C. Verkoulen and M. Voorhoeve, Adaptive workflow — On the interplay between flexibility and support, *Proc. ICEIS* (1999), pp. 353–360.
28. W. M. P. van der Aalst, Generic workflow models: How to handle dynamic change and capture management information? *Proc. Int. Conf. Cooperative Information Systems* (1999), pp. 115–126.
29. J. Mendling, H. A. Reijers and W. M. P. van der Aalst, Seven process modeling guidelines (7PMG), *Inf. Softw. Technol.* **52** (2010) 127–136.
30. A. Agostini and G. D. Michelis, A light workflow management system using simple process models, *J. Comput. Supported Cooperative Work* **9** (2000) 335–363.
31. I. Davies, P. Green, M. Rosemann, M. Indulska and S. Gallo, How do practitioners use conceptual modeling in practice? *Data Knowledge Eng.* **58** (2006) 358–380.
32. C. Ardito *et al.*, Visual mediation mechanisms for collaborative design and development, *Universal Access in HCI* (2011), pp. 3–11.
33. L. G. Tudor, M. J. Muller, T. Dayton and R. W. Root, A participatory design technique for high-level task analysis, critique, and redesign: The CARD method, *Proc. HFES'93* (1993).
34. M. J. Muller, Layered participatory analysis: New developments in the CARD technique (2001).
35. D. Lafrenière, CUTA: A simple, practical, and low-cost approach to task analysis, *Interactions* **3** (1996) 35–39.
36. A. Klær and K. H. Madsen, Participatory analysis of flexibility, *Commun. ACM* **38** (1995) 53–60.
37. K. Halskov and P. Dalsgård, Inspiration card workshops, *Proc. 6th ACM Conf. Designing Interactive Systems. DIS '06* (2006), pp. 2–11.

38. E. Beck, M. Obrist, R. Bernhaupt, M. Tscheligi and T. Cedex, Instant card technique: How and why to apply in user-centered design, *Proc. Particiapatory Design Con.* (2008), pp. 162–165.
39. B. A. Nardi, *A Small Matter of Programming — Perspektives on End-User Computing* (The MIT Press: Cambridge, 1995).
40. Y.-C. Hsu, *The Effects of Visual Versus Verbal Metaphors on Novice and Expert Learners' Performance. Human-Computer Interaction*, Vol. 4 (Springer, Berlin, 2007), pp. 264–269.
41. A. F. Blackwell, The reification of metaphor as a design tool, *ACM Trans. Comput. Human Interaction* **13** (2006) 490–530.
42. A. F. Blackwell, N. Kirsten, K. N. J. Whitley and M. Petre, Cognitive factors in programming with diagrams, *Think. Diag.* (2001), pp. 95–114.
43. P. T. Cox and B. Song, A formal model for component-based software, *Proc. IEEE 2001 Symp. Human Centric Computing Languages and Enviroments* (*HCC'01*) (2001).
44. M. Won, O. Stiemerling and V. Wulf, Component-based approaches to tailorable systems, *End-User Development* (2006), pp. 115–141.
45. F. Daniel, S. Soi, S. Tranquillini and F. Casati, From people to services to UI: Distributed orchestration of user interfaces, *Proc. BPM* (2010), pp. 310–326.
46. M. Costabile, P. Mussio, L. P. Provenza and A. Piccinno, Supporting end users to be co-designers of their tools, *Proc. 2th Int. Symp. End-User Development* (2009), pp. 70–85.
47. L. Strachan, J. Anderson, M. Sneesby and M. Evans, Minimalist user modelling in a complex commercial software system, *User Model. User-Adapted Interaction* **10** (2004) 109–146.
48. T. Hermann, Communicable models for cooperative processes, *7th HCI Int. Design of Computing Systems: Cognitive Considerations* (1997), pp. 285–288.
49. S. S. Hussain, D. Veiel, J. M. Haake and S. Lukosch, Integrating context-enriched explanations und communication in an adaptive collaboration environment, *Groupware: Design, Implementation, und Use: 16th Int. Workshop* (*CRIWG 2010*) (2010).
50. G. Iacucci, V. Pipek, G. Jacucci and K. Kuutti, Continuing design in use of tangible computing environments, *The Good, the Bad and the Irrelevant — The User and the Future of Information and Communication Technologies* (*COST 269 Conference*) (2003), pp. 209–216.
51. J. A. Landay and B. A. Myers, Interactive sketching for the early stages of user interface design, *Proc. SIGCHI Conf. Human Factors in Computing Systems* (1995).
52. J. Grundy and J. Hosking, Supporting generic sketching-based input of diagrams in a domain-specific visual language meta-tool, *Proc. 29th Int. Conf. Software Engineering* (2007).
53. M. Nakagawa, N. Kato, K. Machii and T. Souya, Principles of pen interface design for creative work, *Proc. Second Int. Conf. Document Analysis and Recognition* (1993), pp. 718–721.
54. T. Winograd, From programming environments to environments for designing, *Commun. ACM* **38** (1995) 65–74.
55. D. W. Randall, R. H. R. Harper and M. Rouncefield, *Fieldwork for Design: Theory and Practice* (Springer, London, 2007).
56. F. Kensing, J. Simonsen and K. Bødker, MUST — A method for participatory design, *Proc. Fourth Biennial Conf. Participatory Design* (1996), pp. 129–140.
57. R. B. Yeh, S. R. Klemmer and A. Paepcke, Design and evaluation of an event architecture for paper UIs: Developers create by copying and combining, *Technical Report, Stanford University Computer Science Department* (2007).

58. J. O. Wobbrock, A. D. Wilson and Y. Li, Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes, *Proc. 20th Annual ACM Symp. User Interface Software and ACM* (2007).
59. J. Grundy, J. Hosking, L. Li and N. Liu, Performance engineering of service compositions, *Proc. Int. Workshop of Service-Oriented Software Engineering* (2006), pp. 26–32.
60. G. Fischer, User modelling in human-computer interaction, *User Model. User-Adapted Interaction* **11** (2004) 65–86.
61. V. Pipek, From Tailoring to Appropriation Support: Negotiating Groupware Usage, Ph.D. Thesis, Oulu, Finland, 2005.
62. C. Doerner, S. Draxler, V. Pipek and V. Wulf, End users at the bazaar: Designing next-generation enterprise-resource-planning systems, *IEEE Softw.* **26** (2009) 45–51.
63. C. Doerner, F. Yetim, R. Singh and V. Pipek, SISO — A process design environment for business users, *European Conf. Information Systems* (2010).