

# Business Process Elaboration through Virtual Enactment

Stefan Oppl

Department of Software Science  
Radboud University Nijmegen  
Toernooiveld 212  
6525 EC Nijmegen, NL  
stefan.oppl@jku.at

## ABSTRACT

Validation of business process models under involvement of stakeholders is usually performed by moderated model walkthroughs. We explore the potential of combining these established practices with interactive enactment of executable prototypes that is widely used in UI prototyping and model-based interactive system design. Based on these fields, we develop the concept of process elaboration through scaffolded virtual enactment. The proposed concepts are instantiated in a web-based tool that enables to explore the potential of the approach. An initial exploratory study could confirm that the approach is considered supportive by end users and can be used to elaborate existing process models or develop models of work process from ground up.

## CCS Concepts

- Enterprise Computing → Business Process Management → Business Process Modeling
- Human-centered Computing → Interaction Design → Interaction Design Processes & Methods → Participatory Design

## Keywords

Business process validation; Process enactment; Scaffolding; Business process prototyping.

## 1. INTRODUCTION

The active involvement of operative personnel and stakeholders in the design of business process models and for keeping them current has gained raising attention in the last years [9, 15, 20]. Research has acknowledged that people involved in the actual work processes, which are to be represented in models, are valuable providers of input [40] and can be actively involved in the modeling process [8, 29].

The later phases of process design, however, has been examined less extensively in BPM research. These phases are characterized by the availability of initial process models, which still require to be validated and potentially need to be elaborated to cover all relevant aspects of the real-world process [30]. Validation is usually carried out by chauffeured model walk-throughs [13, 30], i.e., a facilitator presents the model to domain-experts and explores the validity of the represented process information.

Interactive validation through enacting models via a dedicated validation interface enables users to play through the model while

keeping track of the progress through the model. Such approaches are inspired by the idea of user interface prototyping [7, 21] and have been heavily adopted in task-model-based interactive system design [5, 19]. While proposed for BPM research already more than 20 years ago [1], this idea hardly has been examined scientifically ever since then and are only adopted in some commercial tools (e.g., Metasonic Proof - [www.metasonic.de/metasonic-proof](http://www.metasonic.de/metasonic-proof)).

In case of process walk-throughs [13, 30], changes to the model are usually performed immediately, allowing users to assess their proposed modifications directly. Immediate changes are usually not possible if validation is based on model execution. Here, design and runtime are kept separate, enabling modifications to the model only in between validation cycles. This is mainly due to conceptual and technical considerations, as changing a model underlying a process instance during runtime usually has implications on the consistency of the instance information [6, 28].

For validation, however, intermediate inconsistent process states might not have as severe implications as when occurring in production systems. As long as inconsistencies are kept track of and resolved until the end of the validation session, process modifications at runtime are possible, if the currently executed instance can be adapted in a way that leaves it in a state that allows for further execution.

Direct adaptation of process models in enactment-based validation sessions would bring together the advantages of model-based process walk-throughs in terms of immediate reflection of changes, with the advantage of the immediacy of workflow validation via enactment as widely demonstrated in task modeling and UI prototyping. The present work sets out to make a first step towards addressing this potential and introduces a virtual enactment platform for validation and elaboration of process models. At the same time, it aims to bring on of the advantages of facilitated model-walkthroughs to the field of validation through model enactment by offering adaptive support in the validation process, offering prompts on what to consider or pointing at potential further steps to be performed during validation (e.g., as proposed by Herrmann & Loser [11]).

The remainder of this paper is structured as follows: In the next section, we discuss the conceptual background of the present research and summarize the state of related work. In section 3, we describe our conceptual approach to support virtual enactment and derive requirements on interactive system support. The developed technical platform is described in section 4. Section 5 summarizes the findings of an exploratory evaluation of the developed system. We conclude with an account on the current limitation of the platform and future directions of research.

## 2. BACKGROUND & RELATED WORK

The conceptual foundations of the present work can be found in three areas as already briefly mentioned in the introduction. We build upon concepts developed in the area of process elicitation and validation through model walkthroughs. These concepts are augmented with interactive exploration of processes by playing through them, similar to approaches that have been developed and deployed in UI prototyping and task-based interactive system design. Users' needs for support in the enactment and elaboration processes are aimed to be met by adopting the educational concept of scaffolding, which aims at providing adaptive and support depending on users' current situation and needs. We briefly outline prior relevant work in these areas in the following.

### 2.1 Process Elaboration through Enactment

The concept of walkthroughs for exploring the design of a socio-technical system (such as a business process, but also end-user software or enterprise information systems) has been first proposed more than 25 years ago. Polson et al. [27] have proposed to use cognitive walkthroughs to evaluate user interfaces of software. Their approach focusses on individual users and their perception of the socio-technical system. Pinelle & Gutwin [26] transfer this concept to the area of work support systems and aim at examining groupware usability by means of walkthrough techniques. They still focus on the individual users as the main subject in the walkthrough and aim at evaluating existing designs rather than actively engaging walkthrough participants in design activities. Herrmann et al. [14] transfer the concept of walkthrough to collaborative settings and explicitly integrate design activities in the walkthrough process, granting participants an active role in validating and refining a socio-technical system. They draw from earlier work in the area of participatory design [18] and specify the procedures embedded in their approach based on approaches like scenario-based design [16] or contextual design [2]. They base their walkthroughs on graphical representations of a model of the socio-technical system, which serve as an anchor for communication and keep the walkthrough focused [12].

Similar concepts have also been proposed in business process elicitation (e.g., [9, 30]). The proposed approaches, however, focus on knowledge elicitation in models and their transformation to a form processable in BPM activities. They remain vague with respect to the actual procedures in the walkthrough and leave guidance to a facilitator, whose role is not discussed in detail. Hjalmarson et al. [15] explicitly examine the role of facilitators in system analysis and design but focus on identifying generic facilitation strategies rather than describing their activities in detail. We thus mainly draw from the research presented by Herrmann et al. (e.g., [14]) in the following.

The idea of using enactable prototypes of systems for the purpose of walkthroughs has been adopted early in the area of user interface validation [21] and is a common practice in this area ever since then. It was eventually picked up for validating work support systems in the area of task-based interactive system design (e.g., [5, 19]). Sousa et al. [33] propose to combine business-process modeling with task-based user interface design with involvement of stakeholders to create a better fit between their expectations and the interactive system's properties. A similar approach is proposed Sukaviriya et al. [36], who use business process models as the foundation for rapid user interface prototyping.

All these approaches have in common, that they distinguish design-time from run-time in model processing and thus do not

allow to make changes to a prototype, while it is being enacted. More recent research has explored the potential for runtime adaptability of systems (e.g., [6, 10, 28, 31]), making it possible to modify user interface prototypes [10] or whole business process architectures [6, 28, 31] while they are being executed.

We propose to adopt the idea of runtime adaptability of processes to extend the ability of performing design activities during walkthroughs as proposed by [14] to validation and elaboration activities that are based on enactable prototypes of business processes.

Such an approach would require facilitation that goes beyond what is required in model-visualization-centric walkthroughs (as not all aspects of the process are visible all of the time), but also what is used in traditional iterative prototyping processes (as changes to the underlying model can occur during running instances). We hypothesize that the educational concept of scaffolding here can provide the foundation for developing and deploying appropriate support measures.

### 2.2 Scaffolding

Scaffolding is a concept introduced in the field of educational science by Wood et al. [41], adopting a metaphor from construction industry. It refers to an experienced person helping an unexperienced learner to acquire knowledge about a given topic by providing temporary means of support. The support measures, referred to as scaffolds, are deployed until the supported entity (here: users elaborating on process descriptions) can accomplish a given task itself [37]. In order for scaffolds to be acceptable for users and provide added value to them, they need to be appropriated to their current skill level [4].

Scaffolding can take different forms. Based on a meta-study of scaffolding research, Jumaat & Tasir [17] distinguish conceptual scaffolds, meta-cognitive scaffolds, strategic scaffolds, and procedural scaffolds. *Conceptual scaffolds* help learning to decide what to consider to be worth learning. In particular, they can help to prioritize fundamental concepts. *Meta-cognitive scaffolds* guide students in how to approach a learning problem and what to think about when elaborating on a problem. *Strategic scaffolds* suggest alternative ways to tackle problems in learning. Finally, *procedural scaffolds* assist students in using available tools and methods and point them at potentially useful resources. As can be recognized, the different types of scaffold differ in the level of concreteness of the support they provide. The latter types potentially constrain learners' behavior more than the former types, providing stronger guidance, and are consequently are better suited in early phases of the learning process or when contingencies arise [24].

Scaffolding is usually performed by teachers or peers, but can also be provided by means of information technology [3]. This includes interactive systems that try to intervene appropriately in the learning process based on observing learners' behaviors or static intervention rules.

Independently of how scaffolding is implemented, it is always characterized via the presence of three principles that have been identified by van de Pol et al. [37]: The first common principle is *contingency*, which is often referred to as responsiveness or calibrated support. Scaffolds need to be adapted dynamically to the learners' current level of performance. The second principle is *fading*, which refers to the gradual withdrawal of the scaffolding. As learners develop their skills, support becomes less necessary and is reduced over time. This is closely connected to the third principle *transfer of responsibility*. Via fading, responsibility for

the performance of a task is gradually transferred to the learner. The responsibility for learning is transferred when a student takes increasing control about the learning process. The implementation of these principles is based on diagnosis of a learners' need for support, which is usually done by a teacher [34], but also can be implemented in interactive systems [35].

Scaffolds can be implemented via different means, depending on the required support and the context of deployment. Van de Pol et al. [37] (non-exhaustively) list measures such as giving feedback, providing hints, instructing, explaining, modeling (i.e., demonstrating the skill to be acquired) and questioning.

### 3. SCAFFOLDED VIRTUAL ENACTMENT

Based on the conceptual considerations outlined in the introduction and considering the state-of-the-art in research in process model elaboration and scaffolding as described above, we hypothesize that process models can be validated and elaborated by enacting them in an artificial setting (i.e., not situated in a real-world-context impacting actual business cases) and performing changes to model whenever issues are identified. We refer to this process as "virtual enactment" in the following. As is known from facilitated model walkthroughs, domain experts and stakeholders might require support in their model elaboration activities, in particular when they do not have in-depth experiences or expertise in such activities [11, 15]. The importance of dynamically adapting the level of support depending on the level of experience or expertise of the stakeholders is especially stressed in [15] based on empirical evidence. We therefore augment virtual enactment with the educational concept of scaffolding, which inherently requires supportive interventions to be designed and deployed in an adaptive way [37]. The overall conceptual approach presented in this paper thus is referred to as "scaffolded virtual enactment".

#### 3.1 Conceptual Considerations

Virtual enactment as specified here draws from the concept of facilitated model-walkthroughs [13], which are usually carried out in a co-located group setting, bringing together all involved stakeholders at the same place and the same time to collaboratively go through the process. A distributed form of model-walkthroughs can be imagined (and actually has been considered in our own earlier work [38, 39]) but is not subject of the present approach due to the inevitable loss of communication and negotiation potential that would need to be compensated for by further groupware instruments. This in turn could interfere with the core aim of the present work, namely to examine the feasibility of virtual enactment as such, and act as a confounding variable in our studies.

Designing the instrument for co-located synchronous enactment has implications on the necessary support measures. The process fundamentally should be enacted in an actor-centric way, i.e., use the involved process actors as the primary dimension for structuring process enactment. At the same time, all involved stakeholders should have the opportunity of observe the progression through the process across all involved process actors, maintaining an overall bird-eyes-view on what is happening throughout the work process.

Motivated by prototyping research, enactment should allow to go through the work process in an explorative way, as if it were a role-playing game. This implies, that only one path through the process can be enacted at a time, leading to the need to re-enact the process several times to fully explore it. Enactment following the role-play approach requires that stakeholders can focus on these subjects in the process model, whom they also impersonate

in real-world work. This strengthens the point for subject-oriented structuring of process execution. Conceptually separating the behavior of each involved subject and coupling them by acts of communication and/or exchange of information or physical goods should allow to further strengthen the focus on the individual subjects and support the stakeholders in remaining in their roles impersonating the subjects. This can be enabled by a process representation that is similar to S-BPM but adapted to the needs of model elicitation and articulation [23].

Elaboration of the process must be possible during enactment, whenever the need arises, without having to start over the enactment process from the beginning. This avoid losing the context of the current walkthrough and further facilitates that stakeholders can follow-up on their current line of thoughts, if more than one modification should be made. Changes to a process might trigger cascaded need for change, in particular if communication with other process actors is involved (cf. [23]). Elaboration needs to keep track of unsatisfied dependencies (e.g., information expected by a subject to perform its tasks, which is not provided by any other subject) or other issues introduced by local model changes (e.g., dead locks or non-terminating loops). Stakeholders need to be pointed at such issues and need to have the opportunity to resolve them.

To deploy scaffolding for supporting the enactment and elaboration process, the provided scaffolds need to be designed in a way, that allows for situation-specific support that is adaptable by the stakeholders themselves according to their perceived needs. Scaffolds need to be designed for different areas: the process of enactment and process exploration might require guidance or active intervention in particular for inexperienced users. Exploration could further be aided by less invasive scaffolds, such as means to display a graphical representation of the model and the current state of enactment on demand. The elaboration process should be scaffolded in a way that does rely on any modeling skills, as these cannot necessarily be expected from stakeholders. Issues and inconsistencies in the model introduced by local model changes through elaboration can also be pointed out via scaffolds that are dynamically generated based on an analysis of the current state of the model. In any case, stakeholders must have the freedom to ignore scaffolds, dismiss them, and ultimately take responsibility to request support when they consider it necessary.

#### 3.2 Requirements on Tool Support

The conceptual considerations described above have implications on the required tool support for scaffolded virtual enactment. Virtual enactment relies on the availability of a technical instrument that allows to enact given process models and provide the functionality and support measures outlined above. In particular, such an instrument needs to satisfy the following requirements (*EnX* represents a requirement on enactment support, *ElX* represents a requirement on elaboration support, and *ScX* represents a requirement on scaffolding the enactment and elaboration process):

- *En1*: All subjects and their current state in the process need to be accessible to the group of stakeholders enacting the process.
- *En2*: Processes need to be enactable several times in a line, with keeping track of the already explored variants.
- *El1*: Elaboration of process models needs to be possible without canceling the current enactment process, i.e., enactment has to continue at the changed position of the process.

- *EI2*: Elaboration should not require users to manipulate graphical models of the process, but should derive the necessary model modifications from users' descriptions of the local changes to their work situation.
- *EI3*: Issues or inconsistencies introduced in the process by local changes need to be identified and kept track of. Users must have the possibility to fix them in further elaboration steps.
- *Sc1*: The processes of enactment and elaboration need to be scaffolded in a way that allows inexperienced users to work with the tool.
- *Sc2*: Scaffolds need to be provided on different levels of concreteness and invasiveness so that users can adapt scaffolding to their perceived needs of support and level of experience with using the tool.
- *Sc3*: Scaffolds that actively intervene in the enactment or elaboration process should be provided in a way that supports users in understanding the activities the system performs automatically when a scaffold is used.

In the following section, we build upon these requirements and describe a technical system that aims at satisfying them. Fulfillment of some requirements, in particular those related to scaffolding, can only be assessed via user involvement and consequently will be picked up again in the discussion of the exploratory evaluation reported on in section 5.3.

## 4. PLATFORM ARCHITECTURE

Based on the concept and requirements described in the former section, we have developed an online platform for conducting virtual enactment of process models supported by scaffolding measures. In the following, we give an overview about the platform architecture, before we detail on the features of the implemented modules.

### 4.1 Overview

The virtual enactment platform is implemented in a modular way and accessible to users via a web-based interface (<http://adaptivetesting.ce.jku.at/VirtualEnactment>). It is created based on the Vaadin-framework (<http://www.vaadin.org>) that enables to build interactive web-based applications with transparent binding between clients-side and server-side components.

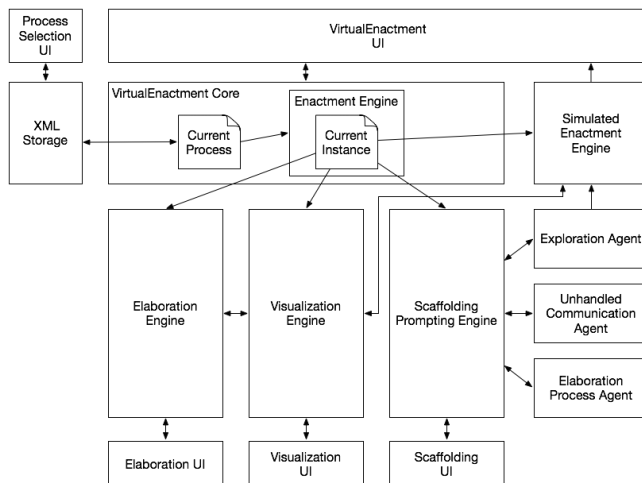


Figure 1: Platform Architecture

Figure 1 gives an overview about the overall architecture of the platform. UI components are shown at the top and bottom of the figure, while functional modules are grouped in the center region.

The *VirtualEnactment Core* provides fundamental workflow execution capabilities that are used for enacting a process model. As such, it acts as the anchor for all other components, which enable elaboration of the currently executed model and provide support to users in this process.

The *Visualization Engine* renders graphical representations based on the current process and can visualize the execution progress of the current instance. Graphical representations are based on the usual S-BPM approach, separating interaction diagram and subject behavior diagrams.

The *Elaboration Engine* allows for changing a process model, while an instance is currently being executed. Retrieving the necessary information is based on prompting. Users can indicate that they consider the currently proposed activity to be inappropriate and are then interactively led through the process of providing the information necessary to make the change to the underlying process model.

The *Simulated Enactment Engine* enables to have the system automatically determine a path to a given target state in the model (across the behavior of all subjects), and enact this path in a user-traceable way (using UI-scripting, i.e., state transitions are reflected on the user interface).

The *Scaffolding Prompting Engine* enables to provide scaffolds to users in a flexible way. Users can dynamically change their required level of support, which is reflected in providing more of less concrete scaffolds. Scaffolds are basically based on textual prompts, but can also provide interactive support measures (such as automatically progressing to a particular state in the model using the simulated enactment engine). The scaffolds themselves are generated by *Scaffolding Agents*, which can focus on different aspects of the modeling and elaboration process. Three different agents are currently provided, which are described in more detail below.

The *XML Storage* component provides functionality to upload a new process model and download altered process models in a proprietary XML format. Users can furthermore select from different sample models or start a new process specification from scratch (as the elaboration engine provides bootstrapping features that allow to define new subjects, their behavior, and their interaction using the same prompting mechanism as deployed for elaboration).

### 4.2 VirtualEnactment Core

The VirtualEnactment Core is the component used for executing a process during virtual enactment. The execution engine consequently is tailored towards this use case. Virtual enactment does not require distributed user interfaces (i.e., only requires one common interface for all participants, visualizing the current states of all subjects at the same time - cf. *EnI*). The current version of the core does not support business object handling, i.e., action states and messages are solely represented by their names. Incoming messages are stored in an input pool, from which they are removed as soon as the respective receive state is triggered.

Figure 2 shows an example of the enactment UI. As can be seen, subject *Secretary* currently is in action state with two potential outcomes (cf. Figure 4 for a visualization of the respective subject behavior). subject *Employee* is on hold in a blocking receive state, and the behavior of subject *Boss* has not yet been started. The

button labeled *Perform step* triggers the shown state to be processed by the workflow engine, the button labeled *I have a problem here* triggers the elaboration engine (cf. section 4.4), while the button labeled *Show behaviour* triggers the visualization engine to display the respective subject behavior diagram.

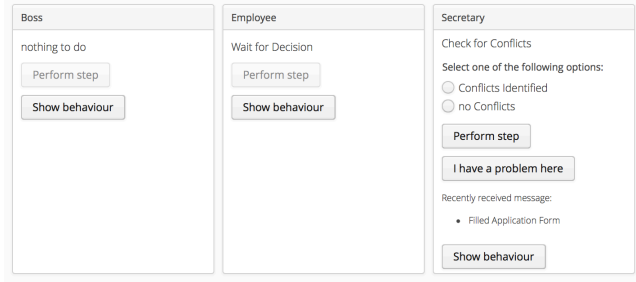


Figure 2: Enactment UI

Elaboration can lead to overall processes that contain inconsistent subject behaviors. One subject's behavior might rely on the availability of a message, which is not currently provided by the envisaged sender. Such messages are added to the envisaged sender's pool of expected messages and can be triggered via the UI as shown in Fig. 3.

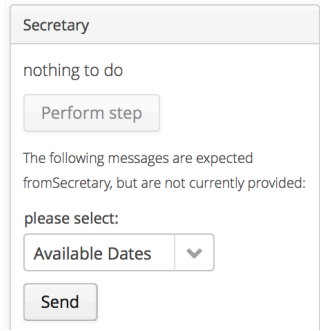


Figure 3: Expected messages in subject UI

In turn, subjects might also provided messages to other subjects, which do not react on these messages in their current behavior. Such messages are added to the envisaged recipient's pool of provided messages (cf. E13). Both, the pool of expected and provided messages is used as a source of information by the elaboration engine. In this way, expected and provided messages can be incorporated in a subject's behavior by adding send and receive states, respectively (cf. section 4.4).

Once an instance of the process is finished, it can be restarted to enact it another time. The new instance takes into account all changes to the process model that might have been made via elaboration during prior enactments. In this way, the process model can gradually be explored in all its variants and be elaborated, where necessary (cf. En2).

### 4.3 Visualization Engine

The visualization engine is used to display subject interaction and subject behavior diagrams. It automatically layouts the diagrams using the GraphViz package REF via the viz.js REF library. The diagrams are dynamically generated upon user request and thus always reflect the current state of the process (cf. En1). This is necessary, as both, subject behaviors and subject interaction might change due to elaboration at any time.

Figure 4 shows an example of a subject behavior diagram. It visualizes the behavior of subject *Secretary* as described above. Already executed states are shown with a bold gray outline. In this

way, the path through the process up to the current state is visualized.

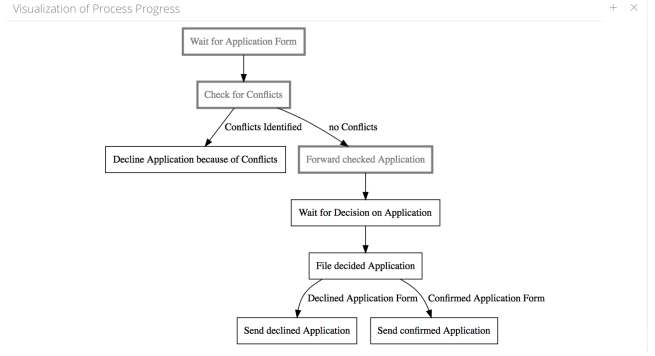


Figure 4: Subject behavior diagram

The visualization engine also provides an interactive mode, in which users can select states or subjects in the currently shown diagrams. This functionality is needed in single use cases of the elaboration engine and the simulated enactment engine as described below.

### 4.4 Elaboration Engine

The elaboration engine allows to alter a process model while an instance of the process is currently being executed. It is anchored in the enactment UI and is always triggered in the context of a particular state. Whenever users consider a state inappropriate to be executed (for whatever reason), the elaboration engine allows to alter process around that particular state (cf. E11).

Elaboration is guided via interactive prompting. Users are not confronted with business process modeling concepts or nomenclature, but can describe what they want to change in the currently enacted work context (cf. E12). Figure 5 shows the sequences of prompts in the interactive elaboration process. Boxes indicate user interaction prompts, while vertical brackets indicate changes made to the process model in the background.

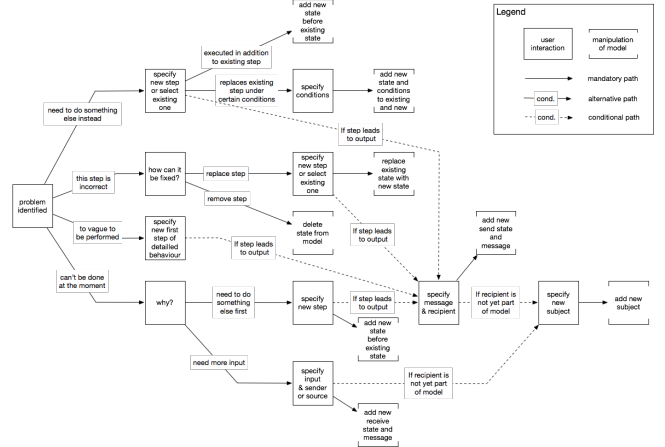


Figure 5: Prompting sequence for elaboration

An example for an interactive prompt is shown in Fig. 6. It shows the user interaction for the element labeled *specify new step or select existing one* in the topmost branch in Fig. 5. Clicking the button labeled *Let me choose from existing steps* would trigger the visualization engine and display the behavior diagram for the respective subject in interactive mode. Alternatively, a new activity can be specified by entering its name in the text field. If the checkbox labeled *This step leads to results I can provide to*

*others* is ticked, the optional path towards element *specify message & recipient* is triggered additionally.

**Figure 6: Example for interactive elaboration prompt**

Wherever necessary, the prompts dynamically adapt to the current state of the process. Fig. 7 shows an example for this feature. It shows the user interaction for *specify message & recipient* as referred to above.

**Figure 7: Specification of messages during elaboration**

In this particular case, the pool of expected messages for the respective subject already contained a message named *Available Dates*. In case the users want to incorporate this message in the subject's behavior, no further input is necessary, as the envisaged recipient of the message has already been specified in the elaboration process, during which the message was defined (via the user interaction element *specify input & sender or source* in the lowermost branch in Fig. 5). If users choose to specify a new message to be provided to others (as shown in Fig. 7), the list of potential recipients is dynamically created from the set of subjects currently contained in the process model. Option *Somebody else* consequently would trigger the option path leading to the addition of a new subject. Option *I do not know who could be interested* creates an anonymous subject, which is shown on the enactment UI. While the behavior of anonymous subjects cannot be elaborated, they still can be used to trigger sending of expected messages (in case the envisaged sender was unknown during elaboration of required input).

The elaboration engine also has process model bootstrapping capabilities, i.e., it can be used to elaborate an initially empty process model. In such cases, the enactment UI offers to add an initial subject. The behavior of this subject can then be elaborated by adding an initial state. Whenever the behavior of a subject is finished in a particular instance, the elaboration engine offers to add an additional state. In this way, process descriptions can be built up from scratch, elaborating the behavior of the initial subject and its interaction requirements in a first process instance, and then gradually refining the model in follow-up instances.

## 4.5 Simulated Enactment Engine

Playing through instances of complex processes repeated times might become tiresome, especially when the initial parts of the process are already agreed upon and elaboration is going on in later parts, which need to be manually navigated to in each new

instance. The simulated enactment engine provides functionality to automate this navigation process and start manual enactment only in the still interesting or questioned part of a process (cf. *ScI*).

The simulated enactment engine searches for a path to the specified target state in the respective subjects' behavior. It then recursively traverses the messages of all encountered receive states, searching for paths to the respective send states in the sending subject's behavior. In this way, a subject-spanning path to the requested target state is compiled, considering both, states to be executed and decisions to be made.

The sequence of steps constituting the path from the current state of the running process instance to the requested target state is then used for UI-scripting. Making use of the Vaadin-framework's server-side push-capabilities, the steps are executed with short delays in between, making it possible for users to follow the simulated enactment process on the UI (cf. *Sc3*). Simulated enactment stops at the requested end state and hands back control to the users for further manual enactment of the currently running instance.

## 4.6 Scaffolding Prompting Engine

The scaffolding prompting engine provides dynamic and user-adaptable scaffolds for different aspects of the exploration and elaboration process. The engine offers an extensible architecture, relying on scaffolding agents to provide the actual scaffolds for a given enactment situation. Scaffolding agents are dynamically registered with the engine and can draw from any source of information (in particular process and instance data). They can choose whether they want to be triggered for providing new scaffolds after each execution step or if a new instance is started. Concrete examples of different scaffolding agents are described below.

The basic form of a scaffold is a text-based prompt, that is displayed in a dedicated area of the UI below the subjects of the current process. Fig. 8 gives an example of the table of scaffolds.

| What to consider:  |              |
|--|--------------|
| Play through your work process in different variants several times to check its appropriateness. | Show Details |
| Secretary contains step "Forward checked Application" that has not yet been explored.            | Show Details |
| Employee contains step "Book Holiday" that has not yet been explored.                            | Show Details |

**Figure 8: Scaffolding prompts**

The slider bar on the left border of the area can be used to adapt the concreteness of scaffolds to be displayed (cf. *Sc2*). Depending on the requested level of concreteness, the engine displays either procedural (most concrete), strategic, meta-cognitive, or conceptual scaffolds (least concrete). Placing the slider at the bottom turns off the display. For the purpose of selecting the appropriate scaffolds to be displayed, scaffolds are grouped together according to their intended effect. Consequently, a scaffolding agent might provide a group of scaffold for a particular intended effect containing a procedural scaffold and a meta-cognitive scaffold, omitting strategic and conceptual scaffolds. In this case, the engine displays the most concrete available scaffold for the level requested by the users. If users requested strategic scaffolds, the meta-cognitive scaffold would consequently be displayed.

A further level of user-control about displayed scaffolds is the ability to dismiss scaffolds. The engine keeps track of dismissed scaffolds and does not display them and less concrete scaffolds of the same group anymore, although they still might be provided by



the scaffolding agents. This avoids annoying users with scaffolds that they deem unhelpful or unnecessary.

The text-based prompts displayed in the table can be detailed in arbitrary ways. The current implementation — aside the basic single-line-prompt-only scaffold — offers a type of scaffolds that can display further information in a pop-up window upon request (cf. Fig. 9) and type of scaffolds that can trigger the simulated enactment engine to take users to that part of the process, which the scaffold suggests to explore further (cf. Fig. 10).

The scaffolding prompting engine is triggered by the process execution engine after each change in the process instance or the underlying process model. The engine informs its registered agents according to their requested update-frequency (per instance or per executed step) and provides them with information about both, the currently used process model and the current instance.

#### 4.6.1 Elaboration Process Agent

The elaboration process agent is a simple agent implementation, which does not provide any dynamically created scaffolds at all. It aims at supporting novice users in using the platform (cf. *ScI*). Consequently, it provides scaffolds introducing the features of the platform distributed over the first few executed instances. While initially users are only asked to explore the process using the execution UI, the agent gradually, whenever a new instance is started, issues scaffolds introducing the visualization UI and the elaboration UI. In this way, users are introduced to the platform's features step-by-step.

#### 4.6.2 Exploration Agent

The exploration agent keeps track about the already executed states contained in the model across all executed instances. In this way, it can provide scaffolds on whether a subject's behavior still has unexplored parts, which could be visited by the users in the current instance (cf. *En2*). The agent always only provides one scaffold per subject, not indicating all unexplored model parts of a subject's behavior at once, but only pointing at one per instance.

Fig. 9 shows a strategic scaffold pointing at an unexplored part of the behavior of subject *Secretary* (behavior shown in Fig. 4). The button labeled *Take me there* triggers the simulated enactment engine, which automatically progresses the current instance to the suggested state, independently of the current state of the instance (assuming the suggested state still can be reached, otherwise a prompt to try again with the next instance is displayed).

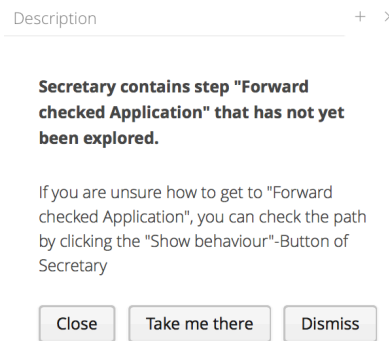


Figure 9: Example for exploration scaffold

#### 4.6.3 Unhandled Communication Agent

The unhandled communication agent offers an example which tries to support the actual elaboration process. It keeps track of the pools of expected and provided messages for each subject and

provides scaffolds that point users at or direct them towards resolving such model issues (cf. *El3*).

Fig. 10 shows a meta-cognitive scaffold for the fact that the subject *Secretary* has expected messages, which it currently does not provide in its behavior. The respective strategic or procedural scaffolds would contain more concrete directions on how to potentially resolve this issue.

As the availability of provided and expected messages might change at any time because of users' elaboration activities, this agent registers not to be updated per completed instance, but per each completed execution or elaboration step.

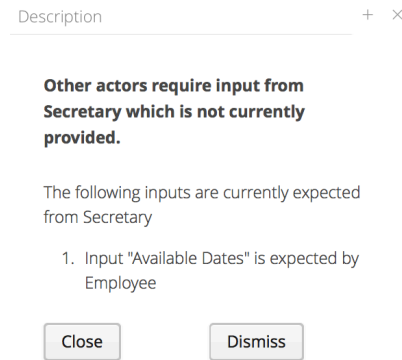


Figure 10: Example for unhandled communication scaffold

### 4.7 XML Storage

The XML Storage component provides process saving and loading capabilities to the platform. It creates a proprietary, yet simple, XML format to create serialized versions of the current process using the XStream library (<http://x-stream.github.io>). Whenever a process instance is completed, the platform offers to download the current version of the process as a XML file. In downloading process versions after each instance, users can keep track of the evolution of the process during elaboration. Whenever users request loading a new process, they can upload files containing data adhering to the same XML format. In its current version, the platform does not provide a server-side process repository. Downloading process versions and re-uploading them later, however, provides a work-around for reverting to earlier versions of a process.

## 5. EXPLORATIVE EVALUATION

The aim of the present paper is to demonstrate the feasibility of using virtual enactment for validation and elaboration of process models. Consequently, the evaluation of the proposed approach and its manifestation in the implemented prototype has focused on exploring its potential and shortcomings to establish a sound foundation for the next design iteration of both, the proposed methodology and the developed tool.

In detail, we have qualitatively examined the understandability and perceived supportiveness of the provided tool, as well as inexperienced users' performance and experiences when elaborating a given process model and when specifying a new model from scratch (i.e., bootstrapping a model without any foundation to elaborate on).

### 5.1 Methodology

The explorative character of the evaluation aims for a controlled setting, in which the influence of contextual factors on modeling, such as individual biases or existing organizational routines, can be reduced when assessing the actual the understandability and

perceived supportiveness of the provided tool. The initial evaluation presented here consequently has been carried out in a quasi-experimental setup, whereas the evaluation of later design iterations of the methodology and tool will require to use case-study-research examining its organizational and individual effects in real-world applications.

The evaluation was carried out with 52 students of business information systems in the second or third year of their bachelor studies (40 males, 12 females, age ranging between 21 and 45). All participants had initial experiences in business process modeling with BPMN [32]. The participants formed 12 groups of 4-5 participants. They were asked to perform 3 elaboration tasks in a given sequence.

The participants were provided with a brief introduction on the fundamental idea of elaboration through enactment and were shown the user interface of the tool. Neither directions for using the system, nor a demonstration of its functionality were provided upfront.

*Task 1* focused on assessing the understandability and perceived supportiveness of the tool. The participants were provided with a simple, yet complete (in terms of depicted alternatives in the work sequence) process description. The process represented how to apply for a vacation in an organization, and contained three subjects — an employee applying for a vacation, a secretary checking for conflicts with other people’s vacations, and a boss responsible for deciding on the application. The participants were asked to play through the process several times to explore all its alternatives. Subsequently, they had to perform two changes to the process by means of elaboration (one behavior extension, one behavior change including an additional message sent to another subject).

*Task 2* asked the participants to elaborate a rudimentary defined process (as could be the result of case-based elicitation of business process models, cf. [22]). Such a task requires that the participants have a subjective understanding of the overall process including its variants. For this reasons, task 2 picked up the scenario underlying task 1 (which the participants had explored previously) and provided them with a model that only contained subject behavior for positively deciding on the application (i.e., no conflicts identified by the secretary, no rejection by the boss). The participants were asked to elaborate the process in a way that it also contained behavior for each subject that allowed rejection of an application.

*Task 3* required the participants to specify a process from scratch. They were given a textual description of a scenario, in which the IT administrator of a SME was responsible for planning and supervising the process of developing a new company website in cooperation with an external web agency. The role of the IT administrator was described in a rather coordinative way, with the CEO of the SME being responsible for making final decisions and further employees being responsible for providing content for the website. The participants were asked to elaborate the process of commissioning the agency with developing the website, including requirements elicitation and internal decision processes. No detailed descriptions of potentially necessary steps were given, but were left to the participants to specify.

All tasks had to be completed by the group without external support (i.e., a facilitator). If stuck, they could ask for support from the supervisor of the study. Overall task completion time was limited to 100 minutes, independently of whether a team could finish all three tasks, to avoid overstraining the participants. They were asked to write down all perceived peculiarities and

problems when completing the tasks. Furthermore, the tool was used in a mode that saved the current state of the present process after each completed instance. In this way, the process of process elaboration could be documented for each task per group. After completing the tasks, the participants were presented the features of the tool as intended to be used by design. For each feature, participants were asked whether they were aware of the tool and considered it supportive.

The raw documentation data were analyzed by means of qualitative content analysis. The saved process models were assessed per task and group to check, which elaboration features the participants had used. The results of the final round of evaluating the supportiveness of the provided features were used to augment the results of the content analysis. The aggregated data were finally interpreted in terms of the evaluation questions described above.

## 5.2 Results

We summarize the results of the explorative study along the evaluation questions stated above. For reasons of space, we refrain from giving detailed accounts on experiences made by single groups, but rather provide an overview about common issues and diagnosed shortcomings of the present prototype implementation.

### 5.2.1 Understandability of the Tool

The intended usage of the tool in general was well understood by all 12 groups involved in the study. The means to enact a process were identified without any support and were used by all groups to play through process instances without observable issues.

The means for visualizing the behavior of the subjects and their interaction was found by all groups and used quite heavily. Some groups traced their way through the process nearly after each step, causing rather heavy load on the server-side visualization engine. While the server could cope with this load and the system remained stable, this was still unanticipated user behavior. After inquiring the reasons in follow-up discussions, it can be attributed to the prior education of the participants in business process modeling, which caused them to mentally track their way through the process and regularly check, whether their mental model matched the actual state of the currently enacted process instance.

The means for elaboration were correctly identified by 10 of the 12 groups — 2 groups did not find out how to start elaboration. In 6 groups, at least once the situation was observed that the participants were unsure of when to trigger the elaboration engine. In all these cases, modifications were to be made to introduce an alternative way of continuing the process (i.e., introducing a conditional split in the control flow). The groups were unsure, whether to trigger elaboration in the state where the condition is to be inserted or in the state that should be complemented by an alternative state (i.e., the state following the future decision state). While the latter variant is the currently implemented way to resolve this issue, people mentally keeping track of the process seem to anticipate the upcoming transition and might want to alter the process already in the transition state.

Further issues were identified with the formulation of several elaboration prompts. Again, potentially attributable to their existing modeling education, all groups at some point started to think in modeling terms and tried to map the prompts to the model changes that would occur in the background. While this is behavior that not necessarily is to be expected from the target group of the tool (stakeholders without any background in modeling), it must not be omitted as irrelevant, as an intended



effect of the tool also to raise awareness for work processes, which in turn could lead to the behavior observed in the study.

The option for manually triggering simulated enactment to progress to a particular state in the process was not found by any group during the study and consequently was not used. The scaffolding prompts were recognized by the users to be available and frequently referred to. Analysis of the log files, however, shows that the option to retrieve extended descriptions or active support (e.g., by simulated enactment) was never used. The slider intended to be used to adapt the concreteness of the scaffold was explicitly considered confusing and was never used beyond initially exploring its effects.

### 5.2.2 Perceived Supportiveness

Perceived supportiveness was examined based on users' feedback on the functional components of the tool. The core functionality provided by the enactment engine was considered supportive for exploring the process. Three groups explicitly noted that it was cumbersome to not be able to undo process steps for local exploration of process variants. Five groups stated that repeatedly going through process parts that already remained stable was an overhead they would prefer to avoid to be able to focus on those parts of the process that they wanted to explore. When the simulated enactment engine was demonstrated, all of them stated that this would have solved their issues, if they had discovered the functionality while completing their tasks (as described above).

The elaboration engine was partially criticized for the used prompts, which were considered to be too unspecific at least in parts by five groups overall. They would have preferred prompts that more explicitly pointed at the process modifications that were to be expected from a specific prompt. As discussed above, these issues — at least in part — can be attributed to the users' prior education in business process modeling. All groups stated that, after an initial phase of accommodation to the prompts, they could cope well with the tasks and handle the elaboration engine appropriately. Three groups noted challenges in tracing cascaded changes over multiple process instances and would have preferred more explicit support in following up on process inconsistencies introduced by a local change to one subject's behavior.

The scaffolding prompts were not considered to be particularly supportive. No group explicitly mentioned the scaffolding prompts as a relevant supportive feature, although observations show, that they were regularly skimmed through while completing the tasks. Users also hardly seemed to recognize (relevant) changes in the list of scaffolds and thus shifted their attention to other parts of the user interface. When being confronted with the ability of scaffolds to actively intervene in the enactment and elaboration process during the demonstration given in the final session, several users stated that this functionality would have been supportive but had not been discovered.

### 5.2.3 Performance and Experiences during Elaboration

Overall, completing the tasks in general took longer than the anticipated 100 minutes. Task completion was checked based on the submitted process models for each group. For task 1, the presence of modifications that were requested were used as a completion criterion. All groups could successfully complete task 1. Task 2 was considered to be successfully completed, if the process contained at least one enactable path that allowed a vacation application to be rejected. Task 2 was successfully completed 11 groups. Task 3 was considered to be completed, if all subjects mentioned in the task description were present,

included at least fundamental behavior descriptions and communication with each other. Only three groups could finish task 3 within the given timeframe. Four of the eight groups, who had started task three had described the behavior of two of the subjects in the task description, whereas the remaining four groups were in their early phases of process description and had only started to specify the behavior of the initially active subject and the sole presence of the other subjects.

The groups that could finish task 2 encountered one conceptual shortcoming of the used version of the elaboration engine, that did not allow to specify message-based conditions for newly introduced alternative behavior. In contrast to manual conditions, message-based conditions react on received messages and select alternative paths accordingly. This lack of functionality was overcome in the study by using manual conditions (i.e., requiring explicit selection of options by the users during enactment). The issue is fixed in the currently deployed version of the tool.

Model bootstrapping (i.e., specifying a model from scratch) was considered cumbersome by four of the seven groups, which could progress beyond the initial phases of model specification in task 3. They in particular criticized the restrictions in adding new subject only when they receive their first message, and would have preferred to add subjects first and then refine their behavior.

The overall feedback on task 2, which in contrast to task 1 resembled a real-world use case and was completed by nearly all groups, was positive. Problems during elaboration were mainly perceived for changes that introduced the need for cascaded further changes. Most problems in this area, however, were attributed to usability issues as discussed above. The underlying concepts of elaboration through virtual enactment were considered useful and viable to be deployed in real-world processes.

## 5.3 Discussion

The results of the study point at several issues that need to be addressed in future revisions of the tool and might have impact on the methodological approach itself.

First, the results clearly indicated that more explicit on-boarding-measures for novice users are needed to introduce them to the features of the tool and demonstrate their use (cf. *Sc1*). This in particular appears to be relevant for the scaffolding prompts, which were hardly used, although they were considered useful when demonstrated after task completion. Refining the functionality of the elaboration process agent in the scaffolding prompting engine to actively intervene in the early phases of tool usage appears to be a viable option that will be explored in future iterations of the tool.

A second area of necessary improvement is motivated by usage of the tool by users who have existing knowledge in business process modeling or might have acquired such knowledge while using the tool (cf. *Sc2*). Such users seem to require more direct ways of interacting with and manipulating the model underlying an elaboration process. One frequently requested feature was the option to permanently display process visualization. The need for more flexible ways of elaboration, i.e., not always being guided by the elaboration prompts, could potentially be satisfied in combination with an optionally permanent model visualization, which could be used to indicate those states that should be modified. Model changes then could be made directly, potentially by means of an interactive editor or modeling-centric prompts (in contrast to the activity-centric prompts used in the current version of the elaboration engine).

Smaller changes that could potentially improve the perceived supportiveness of the tools include an undo-option for enactment steps to allow local exploration of process alternatives without requiring to enact the whole process all over again, and a more differentiated display of the scaffolding prompts, making important or critical scaffolds ones more visible than those pointing at options that users only might want to consider.

Overall, the identified issues discussed above can largely be attributed to shortcomings of the current implementation of the virtual enactment tool. Despite these issues, the study could demonstrate that the approach of elaborating a process through virtual enactment in general is feasible. There is evidence that the approach is usable and supportive for validating and refining existing process models as well as building models of collaborative work processes from scratch.

## 6. CONCLUSION

In this paper, we have introduced the concept of process elaboration through virtual enactment to enable stakeholders to validate and refine existing models of their work processes or specify work processes from ground up. We have deployed the educational approach of scaffolding to support stakeholders in their enactment and elaboration activities. The developed conceptual approaches were manifested in a technical, web-based tool, which was used in an exploratory study examining the feasibility of process elaboration through virtual enactment. We could show that, despite some shortcomings in the deployed implementation, the tool could satisfy the requirements derived from the need for supporting enactment and elaboration processes. The underlying concepts appear to be applicable to real-world scenarios and are considered useful and supportive by end users.

The present work has several limitations. First, some technical shortcomings of the tool have prevented a full exploration of the potential of the underlying concepts. While there is evidence that elaboration during an ongoing enactment process is possible, some more complex cases involving cascaded changes need to be examined more closely. Second, our exploratory study might lack generalizability, as the participants were drawn from a group of students, who already had received training in business process modeling. Future studies will examine the effects of the tool on the main target group, namely organizational stakeholders without any background in business process management and modeling. Third, from a technical point of view, the tool needs to be integrated in the chain of instruments we have proposed for work process articulation and elicitation [23, 25]. This would allow to deploy the concept of virtual enactment in larger-scale use cases that would allow to examine its effects on organizational learning and development processes.

Our future research will focus refining the tool based on the findings of the exploratory study to then deploy it in studies with the main target group to examine the effects of elaboration through virtual enactment for stakeholder without background in modeling.

## 7. ACKNOWLEDGMENTS

This research has been supported by the Austrian Science Fund (FWF): J-3882.

## 8. REFERENCES

- [1] Berztiss, A. 1996. Business Process Prototyping. *Software Methods for Business Reengineering*. Springer. 122–125.
- [2] Beyer, H. and Holtzblatt, K. 1997. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann.
- [3] Bulu, S.T. and Pedersen, S. 2010. Scaffolding middle school students' content knowledge and ill-structured problem solving in a problem-based hypermedia learning environment. *Educational Technology Research and Development*. 58, 5 (2010), 507–529.
- [4] Dennen, V.P. 2004. Cognitive apprenticeship in educational practice: Research on scaffolding, modeling, mentoring, and coaching as instructional strategies. *Handbook of research on educational communications and technology*. 2, (2004), 813–828.
- [5] Dittmar, A., Forbrig, P., Heftberger, S. and Stary, C. 2004. Support for Task Modeling--A "Constructive" Exploration. (2004), 59–76.
- [6] Floch, J., Hallsteinsen, S., Stav, E., Eliassen, F., Lund, K. and Gjorven, E. 2006. Using architecture models for runtime adaptability. *IEEE Software*. 23, 2 (2006), 62–70.
- [7] Floyd, C. 1984. A systematic look at prototyping. *Approaches to prototyping*. Springer. 1–18.
- [8] Forster, S., Pinggera, J. and Weber, B. 2013. Toward an Understanding of the Collaborative Process of Process Modeling. *CAiSE Forum*. (2013), 98–105.
- [9] Front, A., Rieu, D., Santorum, M. and Movahedian, F. 2015. A participative end-user method for multi-perspective business process elicitation and improvement. *Software & Systems Modeling*. (2015), 1–24.
- [10] Hartmann, B., Yu, L., Allison, A., Yang, Y. and Klemmer, S.R. 2008. Design as exploration: creating interface alternatives through parallel authoring and runtime tuning. (2008), 91–100.
- [11] Herrmann, T. and Loser, K.-U. 2013. Facilitating and Prompting of Collaborative Reflection of Process Models. (2013), 17–24.
- [12] Herrmann, T., Hoffmann, M., Kunau, G. and Loser, K.U. 2002. Modelling Cooperative Work: Chances and Risks of Structuring. *Cooperative Systems Design, A Challenge of the Mobility Age. Proceedings of COOP 2002*. (2002), 53–70.
- [13] Herrmann, T., Kunau, G., Loser, K.U. and Menold, N. 2004. Socio-technical walkthrough: designing technology along work processes. *Artful integration: interweaving media, materials and practices. Proceedings of the eighth Conference on Participatory design*. (2004), 132–141.
- [14] Herrmann, T., Loser, K.U. and Jahnke, I. 2007. Sociotechnical walkthrough: a means for knowledge integration. *The Learning Organization*. 14, 5 (2007), 450–464.
- [15] Hjalmarsson, A., Recker, J.C., Rosemann, M. and Lind, M. 2015. Understanding the behavior of workshop facilitators in systems analysis and design projects: Developing theory from process modeling projects. *Communications of the AIS*. 36, 22 (2015), 421–447.
- [16] Holbrook, H. 1990. A scenario-based methodology for conducting requirements elicitation. *ACM SIGSOFT Software Engineering Notes*. 15, 1 (Jan. 1990), 95–104.
- [17] Jumaat, N.F. and Tasir, Z. 2014. Instructional Scaffolding in Online Learning Environment: A Meta-analysis. (2014), 74–77.
- [18] Kensing, F. and Blomberg, J. 1998. Participatory design: Issues and concerns. *Computer Supported Cooperative Work (CSCW)*. 7, 3-4 (1998), 167–185.
- [19] Mori, G., Paternò, F. and Santoro, C. 2002. CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. *IEEE Transactions on Software Engineering*. 28, 9 (2002), 797–813.
- [20] Neumann, G., Sobernig, S. and Aram, M. 2014.

- Evolutionary business information systems. *Business & Information Systems Engineering*. 6, 1 (2014), 33–38.
- [21] Nielsen, J. 1993. Iterative user-interface design. *Computer*. 26, 11 (1993), 32–41.
- [22] Oppl, S. 2015. Articulation of subject-oriented business process models. *Proceedings of S-BPM ONE 2015*. (2015), 1–11.
- [23] Oppl, S. 2016. Articulation of work process models for organizational alignment and informed information system design. *Information & Management*. 53, 5 (Feb. 2016), 591–608.
- [24] Oppl, S. and Hoppenbrouwers, S. 2016. Scaffolding Stakeholder-centric Enterprise Model Articulation. *Proc. of the IFIP WG 8.1 Working Conference on The Practice of Enterprise Modeling*. (2016).
- [25] Oppl, S., Stary, C. and Vogl, S. 2016. Recognition of paper-based conceptual models captured under uncontrolled conditions. *IEEE Transactions on Human-Machine-Systems*. (2016), in press.
- [26] Pinelle, D. and Gutwin, C. 2002. Groupware walkthrough: adding context to groupware usability evaluation. (2002), 455–462.
- [27] Polson, P.G., Lewis, C., Rieman, J. and Wharton, C. 1992. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*. 36, 5 (1992), 741–773.
- [28] Reichert, M. and Dadam, P. 2009. Enabling adaptive process-aware information systems with ADEPT2. *Handbook of Research on Business Process Modeling*. (2009), 173–203.
- [29] Rolland, C., Nurcan, S. and Grosz, G. 1998. A unified framework for modeling cooperative design processes and cooperative business processes. (1998), 376–385.
- [30] Santoro, F.M., Borges, M.R.S. and Pino, J.A. 2010. Acquiring knowledge on business processes from stakeholders' stories. *Advanced engineering informatics*. 24, 2 (Apr. 2010), 138–148.
- [31] Schiffner, S., Rothschädl, T. and Meyer, N. 2014. Towards a Subject-Oriented Evolutionary Business Information System. (2014), 381–388.
- [32] Silver, B. 2009. BPMN Method and Style: A levels-based methodology for BPM process modeling and improvement using BPMN 2.0. *Cody-Cassidy Press, US*. (2009).
- [33] Sousa, K., Mendonça, H., Lievyns, A. and Vanderdonckt, J. 2011. Getting users involved in aligning their needs with business processes models and systems. *Business Process Management Journal*. 17, 5 (2011), 748–786.
- [34] Stender, P. and Kaiser, G. 2015. Scaffolding in complex modelling situations. *ZDM*. 47, 7 (2015), 1255–1267.
- [35] Su, J.-M. 2015. A self-regulated learning tutor to adaptively scaffold the personalized learning: A study on learning outcome for grade 8 Mathematics. (2015), 376–380.
- [36] Sukaviriya, N., Sinha, V., Ramachandra, T., Mani, S. and Stolze, M. 2007. User-centered design and business process modeling: cross road in rapid prototyping tools. (2007), 165–178.
- [37] Van de Pol, J., Volman, M. and Beishuizen, J. 2010. Scaffolding in teacher--student interaction: A decade of research. *Educational Psychology Review*. 22, 3 (2010), 271–296.
- [38] Wachholder, D. and Oppl, S. 2014. Interactive Coupling of Process Models: A Distributed Tabletop Approach to Collaborative Modeling. *ECCE '14: Proceedings of the 2014 European Conference on Cognitive Ergonomics*. (Sep. 2014), 1–8.
- [39] Wachholder, D. and Oppl, S. 2012. Stakeholder-Driven Collaborative Modeling of Subject-Oriented Business Processes. *S-BPM ONE - Scientific Research*. C. Stary, ed. Springer. 145–162.
- [40] Weske, M. 2010. *Business process management: concepts, languages, architectures*. Springer.
- [41] Wood, D., Bruner, J.S. and Ross, G. 1976. The role of tutoring in problem solving. *Journal of child psychology and psychiatry*. 17, 2 (1976), 89–100.