

5

Acting on Work Designs: Providing Support for Validation and Implementation of Envisioned Changes

This chapter introduces methods to enable to act on the results of articulation and alignment processes. As such, it outlines paths towards sharing and anchoring new work practices in organizations. The proposed methods allow to qualitatively validate novel work practices and to agree on strategies for their wider roll-out. Drawing from current practice-oriented research in organizational development, Computer Supported Collaborative Work (CSCW), and knowledge management, this chapter also includes an account on supporting technology that could be used to aid method deployment.

5.1 Creating Executable Models Through Scaffolding Articulation and Alignment

Articulation and alignment of knowledge that guides collaboration in work processes has been addressed in nearly every knowledge management approach proposed over the last decades. As already described in earlier chapters, “articulation” refers to the process of encoding individual mental models (Pirnay-Dummer and Lachner 2008) about a particular

work process in a tangible form (“artifacts”) that enables reflection (Roberts 2009; Prilla 2015) and sharing (Arias and Fischer 2000). Processes of “alignment” consequently build upon these artifacts and are used in collaborative settings to alter and extend the individuals’ mental models. This enables them to operatively work together (Convertino et al. 2008; Oppl 2016) and modify the artifacts accordingly to represent the aligned understanding.

The representation of mental models in externalized artifacts is often approached via creating diagrammatic conceptual models. The capability to use conceptual modeling for the purpose of articulation and alignment must, however, not be assumed to be present for all participating actors (Arias et al. 2000; Hjalmarsson et al. 2015). Inexperienced participants require more support and guidance than experienced ones. However, the latter might be hampered by an enforcement of modeling structures of procedures, leading to less articulation and alignment activities (Franco and Rouwette 2011).

The topic of how to facilitate the development of skills in conceptual modeling in organizational research has been addressed as early as in the 1960s, when Morris (1967) stated that “if one grants that modeling is and, for greatest effectiveness, probably ought to be, an intuitive process for the experienced, then the interesting question becomes the pedagogical problem of how to develop this intuition”. This question has also been picked up in enterprise and work modeling as the discipline continued to mature (Sandkuhl et al. 2014), and has moved away from being considered an “art” that requires “intuition,” to a more scientifically grounded discipline (Willemain 1995).

In recent years, literature recognizes a trend towards a strong and active involvement of stakeholders in the modeling process (Tavella and Papadopoulos 2014) (Recker et al. 2012), who are usually not formally educated in the modeling skills (Powell and Willemain 2007). Models are considered to act as boundary objects (Franco 2013) that enable people who collaborate within organizational structures to articulate and align the understanding of their work systems (Briggs et al. 2013). Research in this domain has focused on how to facilitate modeling activities under involvement of such “novice modelers” (Tavella and Papadopoulos 2014)

towards the ends of generating models appropriate for the respective aims of modeling (Hjalmarsson et al. 2015). In contrast, the question of how to support the development of skills to work with, and on the basis of, conceptual models for this group of people, who usually do not have the opportunity to dedicate effort and time to formal modeling education, has hardly been a subject of research. The potential added value of such skills for this group, however, has been recognized repeatedly over the last decades in terms of pursuing a deeper understanding of the domain and phenomenon being subject of modeling (Mayer 1989; Davies et al. 2006; Hjalmarsson et al. 2015).

5.1.1 Scaffolding

Scaffolding is a concept introduced in the field of educational tutoring by Wood et al. (1976). It originally refers to having an experienced person help an inexperienced learner to acquire knowledge about a particular topic. Scaffolding is a metaphor adopted from construction industry and refers to a temporary means of support that is present until the entity supported by scaffolds (here: an actor participating in conceptual modeling) can accomplish a given task independently (Van de Pol et al. 2010). It is usually motivated by keeping actors in their “zone of proximal development” (ZPD) during a learning process (Vygotsky 1978), that is, putting them in a situation, which is challenging, yet attainable, to them. In order for scaffolds to be acceptable for actors and provide added value to them, they need to be appropriated to their current skill level (Dennen 2004).

Scaffolding can take different forms. Based on a meta-study of scaffolding research Jumaat and Tasir (2014) distinguish conceptual scaffolds, procedural scaffolds, metacognitive scaffolds, and strategic scaffolds. *Conceptual scaffolds* help learn to decide what to consider to be worth learning. In particular, they can help to prioritize fundamental concepts. *Procedural scaffolds* assist students in using available tools and methods and point them at potentially useful resources. *Strategic scaffolds* suggest alternative ways to tackle problems in learning. Finally, *metacognitive scaffolds* guide students in how to approach a learning problem and what to

think about when elaborating on a problem. Orthogonally to these categories, Bulu and Pedersen (2010) identify differences in the sources of scaffolding. *Scaffolds provided by teachers* are considered the original form of scaffolding. *Scaffolds provided in interactions among learning peers* refer to the phenomenon that scaffolding can arise from the collective knowledge of a learning group. Scaffolds can also be provided as *textual or graphical representations*, similar to a manual. *Technology-driven scaffolding* uses (information) technology to provide scaffolds. This includes interactive systems that try to intervene appropriately in the learning process based on observing learners' behaviors or static intervention rules.

Independently of which form of scaffolding is pursued, it is always characterized via the presence of three principles that have been identified by Van de Pol et al. (2010): 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 decreased 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 of the learning process. The implementation of these principles is based on *diagnosis* of a learner's need for support, which is usually done by a teacher (Stender and Kaiser 2015), but also can be implemented in interactive systems (Su 2015).

On an operative level, scaffolding is implemented via different means. Van de Pol et al. (2010) list a (non-exhaustive) set of measures such as giving feedback, providing hints, instructing, explaining, modeling (i.e., demonstrating the skill to be acquired), and questioning. They differ in their depth of intervention and the reduction of freedom in students' learning processes. How to appropriately select and implement scaffolding as interventions in the learning process is disputed (ibid.). The described categories and means thus should be considered a framework for observing and designing learning settings, rather than attributing them any normative value.

5.1.2 Scaffolds for Stakeholder-Centric Work Modeling

Our review of related work has shown that, while scaffolding has already implicitly and, to some extent, explicitly been deployed in the field of conceptual modeling, a structured approach to describe and design scaffolding in modeling activities is not available. As argued earlier, augmenting the design of stakeholder-centric modeling activities with a scaffolding perspective could help improve the understanding and creation of enterprise models in the target group. In the following, we therefore review scaffolding approaches proposed in other disciplines, which require skills similar to stakeholder-centric conceptual modeling, in particular with respect to articulation of abstract concepts describing real-world phenomena (Frederiks and van der Weide 2006) and the support of developing a common understanding about these phenomena (Prusak et al. 2012). Based on these approaches, we develop a framework for scaffolding in enterprise modeling, which constitutes our nascent design theory.

5.1.2.1 Scaffolding the Articulation of Models

The process of articulating abstract concepts, being a main activity in conceptual modeling, has been widely examined regarding potential scaffolding support in the field of mathematical and science education.

Ozmantar and Roper (2004) consider teacher interventions as the major means of scaffolding (in the context of mathematical abstraction problems). Their study focuses on examining the activities of the person providing scaffolds. They identify three major facets that they could observe. First, they observed that scaffolding strategies were chosen in an ad-hoc manner based on continuous monitoring and analyzing the actors' performance. Which scaffold is appropriate in which situation cannot be determined ex-ante. It requires continuous monitoring of the learning process to check whether a provided scaffold achieves the intended effect and allow for adapting one's scaffolding strategy. Second, they identify a major means of scaffolding to provide metacognitive scaffolds by organizing the main goal of the learning activ-

ity into hierarchical sub-goals. Third, they could observe fading and transfer of responsibility to take place when models went beyond their initial construction and had begun to stabilize via consolidation activities.

Land and Zembal-Saul (2003) examine how reflection and articulation processes on scientific explanations can be supported by scaffolding. This focus is conceptually close to what other authors refer to as conceptual modeling. They examine means to scaffold reflection and articulation on a longer-term time scale and focus on means of scaffolding via peers. Their scaffolds are deployed via a software platform and are mainly metacognitive, based on task-specific prompting. They could show that their design was useful to learners and led to sophisticated explanations, indicating the construction of elaborate abstractions. They, however, found that the utility of “static” scaffolds as provided by their platform was dependent on the background knowledge of the learners. They thus suggest combining their approach with human instruction that provides more explicit scaffolding, especially for novices.

Stender and Kaiser (2015) discuss the importance of scaffolding the process of developing mathematical models related to real-world problems and validate their appropriateness for problem-solving. Rather than describing concrete scaffolds, they focus on the diagnosis of student needs and fading support measures to facilitate independent problem-solving by students. Based on existing research on adaptive teacher interventions, they identify the invasiveness of different types of scaffolding in terms of restricting student’s freedom of choice on action. Their empirical results allow them to suggest scaffolding interventions in the model articulation process to facilitate problem solving. Their results indicate, among others, the usefulness of decomposition of the modeling task, availability of model simulation tools, and referring to existing knowledge via metacognitive scaffolds.

5.1.2.2 Scaffolding Argumentative Collaboration for Alignment

Several authors have also examined how to scaffold collaborative articulation, in particular with focus on peer facilitation of argumentative processes.

Abdu et al. (2015) examine how the process of peer facilitation can be supported through scaffolding with whole-group interventions in classroom settings. Their focus consequently is on argumentative design that should prevent unguided model creation. They propose to provide strategic scaffolds by demonstrating solution paths upfront, before peer interaction starts. Furthermore, they establish explicit prompting practices for peer collaboration to establish collective responsibility for the learning process.

Chin and Osborne (2010) show how discursive interaction (i.e., argumentation) can be supported by scaffolding in science education. They propose to provide question prompts for enabling peers to ask questions that allow exploring a problem or a proposed solution in depth (King and Rosenshine 1993). They also suggest providing conceptual scaffolds in the form of additional resources on the topic of interaction and procedural scaffolds in the form of guiding structures (such as writing stems or diagram templates). Finally, the authors propose to work with heterogeneous groups, where participants have different views, to facilitate confronting argumentation (Prusak et al. 2012).

Ge and Land (2004) focus on the development of domain-specific question prompts to scaffold problem-solving in peer interaction settings. They establish guidelines for designing such scaffolds that are based on a combination of generic discursive prompting (King and Rosenshine 1993) and domain-specific prompts that they claim should be developed in cooperation with domain experts. They also suggest structuring a discourse via explicit assignment of roles to learners, which they should take in their interaction.

Vogel et al. (2017) present a meta-study on how collaboration scripts can be used for scaffolding in IT-supported learning environments. Collaborations scripts (Kobbe et al. 2007) are strategic scaffolds that specify a sequence of learning activities to be completed to achieve the aim of a particular task. They found that collaboration scripts have positive effects on the acquisition of domain-specific knowledge in relation to the task and collaboration skills in general. They claim that repeated participation and practice in activities supported through scaffolds by collaboration scripts leads to an internalization of the required performative knowledge, and gradually allows withdrawing the script guidance while

still maintaining the developed problem-solving strategies, not only in terms of collaboration, but also regarding domain-specific skills.

5.1.3 A Framework for Scaffolding Model Articulation and Alignment

Backed by the results of prior research outlined earlier, we hypothesize that scaffolds can be developed for the purpose of skill development in stakeholder-centric modeling settings. The modeling process in this context is considered a process of articulation. It is indivisibly embedded in its social context that requires viewing conceptual modeling as a process of co-construction, ultimately leading to a shared understanding about the topic of modeling among the participating actors. Consequently, a conceptual model always only can represent the agreed upon abstractions of the perceived real-world phenomena considered relevant by the participants. Its value is further determined by the chosen representational system that needs to be selected according to the intended purpose, that is, goal of modeling.

Following this understanding, modeling approaches from a scaffolding perspective need to address the following meta-requirements (Walls et al. 1992): (A1) provide scaffolds for the level of model representation (i.e., encoding abstractions in an external representation), (A2) provide scaffolds for the level of model articulation (i.e., developing an understanding about the real-world phenomenon that is the topic of modeling), and (A3) provide scaffolds for the level of collaborative model alignment (i.e., the process of mutually supporting the development of a shared understanding about the topic of modeling and the modeling process itself).

Furthermore, in order to allow for contingency, fading, and transfer of responsibility, (A4) scaffolds need to be provided with different degrees of invasiveness to allow to adapt modeling support to the current needs of the modelers.

Based on these requirements, we draw from the results of our literature review in the following and propose a meta-design (Walls et al. 1992) in the form of a scaffolding framework which should support the process of enterprise modeling on all three levels identified earlier. The framework is visualized in Fig. 5.1.

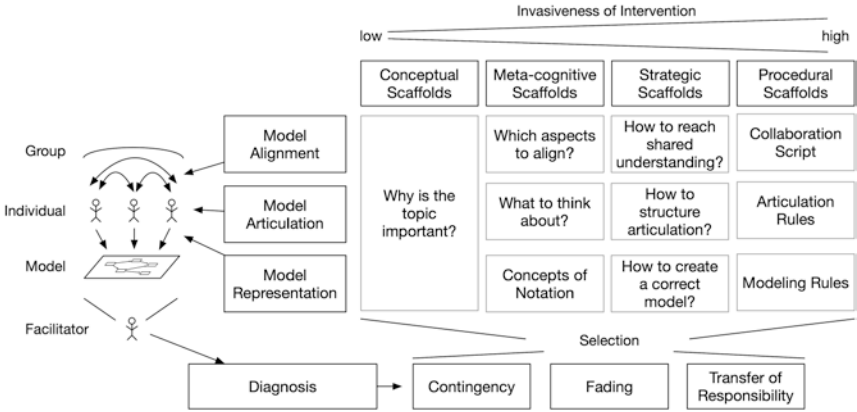


Fig. 5.1 Dimensions of scaffolding during work modeling

The fundamental constituents of the framework are visualized on the top, bottom, and left margin of Fig. 5.1. Its starting point can be found on the left, where our conceptualization of enterprise modeling being activities of a group of actors to create a common conceptual model is shown. As identified earlier, this requires performing model representation, model articulation, and model alignment activities, and is usually supported by a facilitator.

The top margin of Fig. 5.1 structures different types of scaffolds (Van de Pol et al. 2010) according to their invasiveness of intervention (Stender and Kaiser 2015) (cf. A4). Depending on the diagnosis of current needs of the group of stakeholders engaged in modeling, the facilitator is deploying different types of scaffolds following the principles of scaffolding visualized at the bottom margin of Fig. 5.1. The more responsibility is transferred to the modelers, the more support measures are faded out, and scaffolds are deployed (if any) that are less invasive. In case of contingency, the facilitator is free to temporarily fade in stronger support to keep the modelers oriented towards the aim of modeling.

The center of Fig. 5.1 shows the aspects of modeling that should be addressed by different types of scaffolds for model representation (cf. A1), articulation (cf. A2), and alignment (cf. A3). In general, conceptual scaffolds, independently of the addressed level, motivate the topic of model-

ing, show its relevance, and allow to validate the model with respect to their appropriateness in real-world use. Metacognitive scaffolds support the understanding of the structure of the modeling task and indicate how to conceptualize the real-world phenomenon. On the level of model alignment, metacognitive scaffolds indicate which aspects of a model are subject to alignment (i.e., interfaces between model parts, in contrast to those aspects that remain in the responsibility of individual modelers). Strategic and procedural scaffolds aim at supporting the modeling process itself, either by showing potential behavioral strategies in the first case, or providing stricter guidance in the second case. On the level of model representation, such scaffolds focus on syntactic aspects of modeling, whereas articulation and alignment focus on semantic aspects.

Scaffolds of either form and on either level can be delivered via different channels. They can be provided by procedural guidance or by artifacts (static or interactive media) designed to mediate the modeling process. Procedural guidance can be provided by a human facilitator or an IT-based system, in case the latter is capable of monitoring the modeling progress and analyzing the challenges participants currently face on their individual skill level. Procedural guidance by humans can be provided by expert facilitators or peers, when the latter are provided with further scaffolds that guide the facilitation process itself. Designed artifacts can be provided for domain-specific support and for collaboration support, whereas in both cases their value is determined by an anticipated fit between the skill level of the addressed actors and the support provided by the artifact. As this fit usually cannot be taken for granted, pre-designed scaffolds are usually combined with a form of procedural guidance.

Figure 5.2 gives examples of these different types of scaffolds, distinguishing between different sources of scaffolding. The examples are taken from research cited earlier.

The examples should be considered a non-tentative overview about how the different forms of scaffolding can be provided via different delivery channels. They are deliberately not assigned to the different levels of support indicated in Fig. 5.1 (model representation, model articulation, model alignment), as existing literature does not distinguish these levels.

	Conceptual Scaffolds	Meta-cognitive Scaffolds	Strategic Scaffolds	Procedural Scaffolds
Facilitator	Motivation of Conceptual Focus	Prompting	Behavioural Modeling*	Instruction
Other Modelers	Confrontation with Other's Aims	Confrontation with other's Models	Behavioural Modeling*	Collaboration script
Static Medium (Text, Graphics)	Description of Relevant Concepts	Prompting Templates	Structural or Behavioural Templates	Step-by-Step Description
Interactive Medium	Interactive Validation of Model	Interactive Prompts	Interactive Portfolio	Interactive Guidance

* here referring to „acting as a model on how to solve the problem“

Fig. 5.2 Examples of different forms of scaffolds for work modeling

5.1.4 Scaffolding Articulation and Alignment in CoMPArE/WP

The proposed framework can be used to augment the CoMPArE/WP (Collaborative Model Articulation and Elicitation of Work Processes) method (as described in Chap. 4), which explicitly aims at supporting articulation and alignment of stakeholders' views on their contributions to enterprise processes and the collaboration necessary to implement them. The method follows a multi-step modeling approach, in which participants first collaboratively create a concept map to agree on the notions used to refer to the relevant aspects of their work, then individually model their views on their own contributions and interfaces with others, and finally consolidate these models in a discursive way to create an agreed-upon representation of the overall work process. If modeling rules are adhered to, the resulting models are technically interpretable by a workflow engine, and in this way can be validated through simulated execution (Oppl and Alexopoulou 2016).

CoMPArE/WP offers support measures to enable modeling by stakeholders who do not have any prior knowledge in conceptual modeling. These support measures are briefly described in the following and then

classified using the proposed framework (cf. Fig. 5.3). The *global multi-step modeling procedure* is introduced by a facilitator, who is expected to be trained when implementing the method. The participants are also provided with a *one-page written/graphical summary of the global procedure*. The modeling notations for individual articulation and collaborative alignment are pre-specified and are provided via *cardboard model elements* that follow a coloring scheme encoding the semantic elements of the used modeling language. The same coloring scheme is used in *poster-sized printed templates* that indicate the expected model layout that needs to be adhered to in order for the results to be unambiguously interpretable via technical means. Printed *model examples* are provided for reference in case of uncertainty on how to use the model elements or their semantics. The participants have access to *written descriptions of the modeling rules* for each step. In the course of the workshop, the facilitator observes individual model articulation and provides *role-specific prompts* to aid model development. If necessary, the facilitator *demonstrates model development using an example*. During consolidation, the participants are expected to contribute their individual models and support the *identification of model*

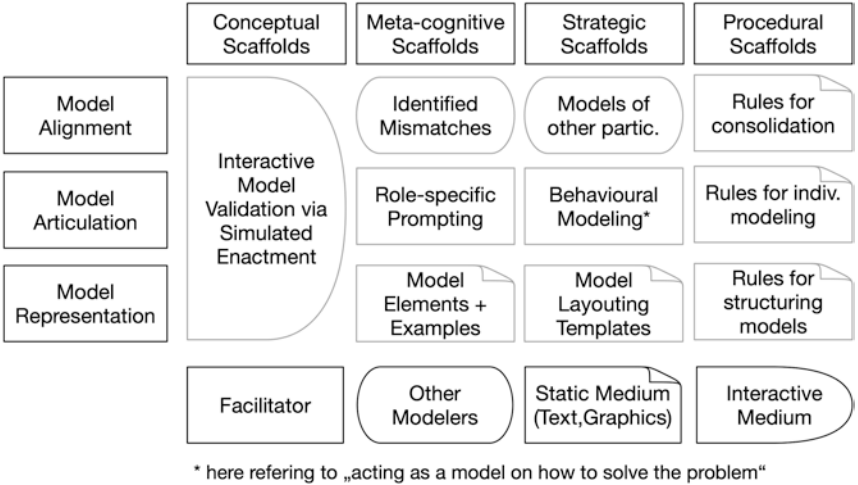


Fig. 5.3 Scaffolds deployed in CoMPARe/WP (references indicate the foundation for design)

parts that indicate divergent views on how to collaborate. The identification process is supported via the *contributed models of all participants* that should contain semantically identical model elements in case of agreement on how to collaborate. The resulting model is transformed into an executable version by means of image recognition (Oppl 2015) and *interactively validated via simulated enactment*, enabling participants to identify inadequacies of the developed model.

5.1.5 Example

Applying scaffolding to CoMPArE/WP leads to observable effects caused by how the facilitators guided model representation and alignment. Figure 5.4 shows a *model layout template* used as a strategic scaffold for model representation for consolidation. The three photos in Fig. 5.4

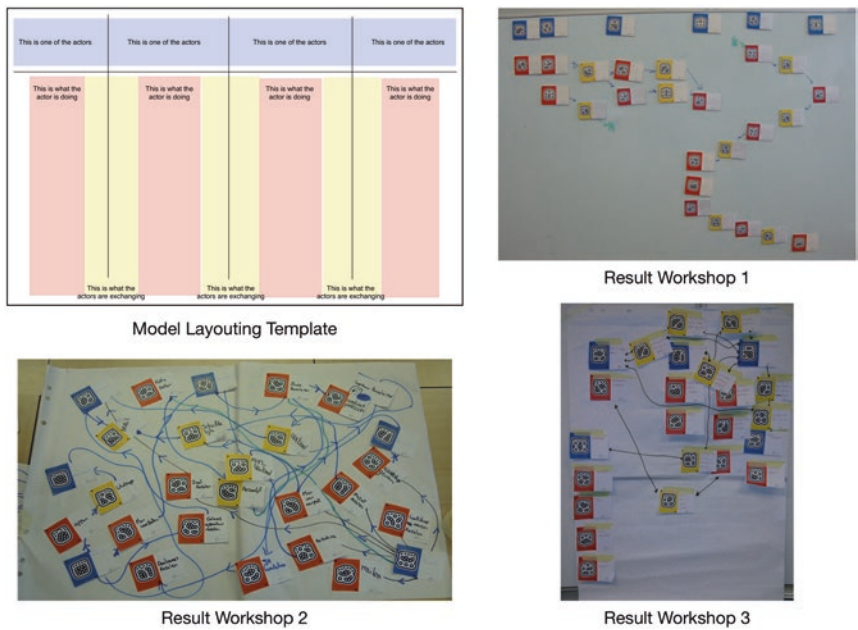


Fig. 5.4 Top left: model layout template; top right and bottom: modeling results of workshops

illustrate the different results of consolidation. On the representation level, the aim is to resemble the layout indicated in the template (blue elements on top, red elements aligned below in lanes, yellow elements placed between lanes). On the alignment level, participants themselves should discover problems in the depicted process (e.g., non-matching communication expectations) and resolve them.

The facilitator in *workshop 1* deployed model representation scaffolds on a strategic and procedural level. Scaffolds for model articulation and alignment were not used. *Fading*, *transfer of responsibility*, or *contingency* could not be observed. This resulted in a syntactically correct model but led to little involvement of the stakeholders in modeling and articulation and no observable alignment activities. The facilitator in *workshop 2* introduced the global multi-step modeling procedure as a high-level procedural scaffold and provided the participants with metacognitive scaffolds on representation, articulation, and alignment. *Contingency* could not be observed, although participants showed signs of being overwhelmed with the task. The facilitator rather *shifted full responsibility* to the participants after an initial deployment of the metacognitive scaffolds. While high involvement of all participants in articulation could be observed, participation was declining during alignment, and led to a syntactically incorrect modeling result, with semantic deviations from the proposed modeling language.

The facilitator in *workshop 3* actively implemented *contingency* and *fading*. The facilitator started with strategic scaffolds for model representation and articulation, briefly provided procedural scaffolds at the start of the consolidation step, and provided metacognitive scaffolds in case of contingency. The observed modeling process continuously showed high involvement in articulation and alignment, with deviations from the proposed modeling notation in the final modeling result.

This example has shown that scaffolding plays a crucial role in developing models that are both adequate representations of the stakeholder's mental models and syntactically correct descriptions of a work procedure that can be processed further for validation and eventually put to practice.

5.2 Participatory Enactment Support Instrument

Once novel work designs are initially represented as syntactically correct and executable process models, they still require to be validated and potentially need to be elaborated to cover all relevant aspects of the real-world process (Santoro et al. 2010). Validation is usually carried out by chauffeured model walk-throughs (Santoro et al. 2010; Herrmann et al. 2004)—a facilitator presents the model to domain experts and explores the validity of the represented process information—or by simulation games—whereby a process is usually played through in a collaborative setting and potential improvement are identified (Smeds and Alvesalo 2003).

Such simulation games, however, are rarely supported by information systems. They rather take place in a social setting that resembles a simplified version of the actual work environment. Interactive validation through enacting models via a dedicated validation information system, nevertheless, could enable users to play through the model while keeping track of the progress through the model, thus relating work experiences more directly to the properties of an underlying process model. Such approaches are inspired by the idea of user interface prototyping (Floyd 1984; Nielsen 1993) and have been heavily adopted in task-model-based interactive system design (Dittmar et al. 2004; Mori et al. 2002). While proposed for BPM already more than 20 years ago (Berztiss 1996), this idea hardly has been examined scientifically since then, and is only adopted in some commercial tools.

In case of process walk-throughs (Santoro et al. 2010; Herrmann et al. 2004), changes to the work process 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 the model of a process instance during runtime usually has implications on

the consistency of the instance information (Floch et al. 2006; Reichert and Dadam 2009).

Validation, in contrast to process execution in production systems, does not lead to actual output. Continuing execution despite missing information or inconsistencies among different process parts thus is an option here, if the validation environment keeps track of such issues and allows manually circumventing or resolving them. Once a process change prevents further execution of an instance, validation can still continue by re-starting an instance.

Direct adaptation of process models in enactment-based validation sessions brings 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. We here introduce a virtual enactment platform for validation and elaboration of process models that addresses this challenge. At the same time, it aims to bring the advantages of facilitated model walk-throughs 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 and Loser 2013).

5.2.1 Background: Process Walk-Throughs and Enacted Prototypes

The concept of walk-throughs for exploring the design of a socio-technical system (such as a business process, but also end-user software or enterprise information systems) was first proposed more than 25 years ago. Polson et al. (1992) have proposed to use cognitive walk-throughs to evaluate user interfaces of software. Their approach focuses on individual users and their perception of the socio-technical system. Pinelle and Gutwin (2002) transfer this concept to the area of work support systems and aim at examining groupware usability by means of walk-through techniques. They still focus on the individual users as the main subject in the walk-through and aim at evaluating existing designs rather than

actively engaging walk-through participants in design activities. Herrmann et al. (2007) transfer the concept of walk-through to collaborative settings and explicitly integrate design activities in the walk-through 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 (Kensing and Blomberg 1998) and specify the procedures embedded in their approach based on approaches like scenario-based design (Holbrook 1990) or contextual design (Beyer and Holtzblatt 1997). They base their walk-throughs on graphical representations of a model of the socio-technical system, which serve as an anchor for communication and keep the walk-through focused (Herrmann et al. 2002).

Similar concepts have also been proposed for business process elicitation (e.g., Santoro et al. 2010; Front et al. 2017). The proposed approaches, however, focus on knowledge elicitation in models and their transformation to a format along BPM activities that can be processed. They remain vague with respect to the actual procedures in the walk-through and leave guidance to a facilitator, whose role is not discussed in detail. This lack of methodological depth is addressed by Caporale (2016), who proposes an approach in which process model constructs are derived from natural language process descriptions, enabling stakeholders to describe their work using familiar constructs. Hjalmarsson et al. (2015) 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.

The idea of using prototypes of systems that can be enacted for the purpose of walk-throughs has been adopted early in the area of user interface validation (Nielsen 1993) 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., Dittmar et al. 2004; Mori et al. 2002). Sousa et al. (2011) propose to combine business-process modeling with task-based user interface design, with involvement of actors to create a better fit between their expectations and the interactive system's properties. A similar approach is proposed by Sukaviriya et al. (2007), 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 runtime 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., Hartmann et al. 2008; Floch et al. 2006; Reichert and Dadam 2009; Schiffner et al. 2014), making it possible to modify user interface prototypes (Hartmann et al. 2008) or whole business process architectures (Floch et al. 2006; Reichert and Dadam 2009; Schiffner et al. 2014) while they are being executed.

We adopt the idea of runtime adaptability of processes to extend the ability of performing design activities in the course of walk-throughs, as proposed by Herrmann et al. (2007), to validation and elaboration activities that are based on prototypes of business processes that can be enacted. Such an approach requires facilitation beyond what is required in model-visualization-centric walk-throughs (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 again adopt the educational concept of scaffolding, as presented earlier, to provide the foundation for developing and deploying appropriate support measures.

5.2.2 Implications of Enacting Dynamically Changeable Prototypes

During participatory enactment, the actors go through the work process step by step. For each step, the responsible actor assesses

- whether the step is correct and described in sufficient detail, and
- whether the next step is the only possible way to progress or if there are alternative ways of continuing with the work process. This can refer to alternative options of progressing, optional activities, or activities that have been omitted in the original model

If any on these assessments lead to the need for changes in the process, these changes would be incorporated in the model underlying enactment

immediately. Changes can have different effects that might trigger the need for further changes in the behavior of other actors (here represented in the work models as subjects, as introduced in Chap. 3) or in the overall process. Potential changes in ascending order with respect to their impact on the overall process are:

1. adding, altering, or removing a subject's activities
2. shifting activities from one subject to another
3. altering the sequence of information exchange between subjects
4. adding or removing information required from or provided to another subject
5. involving a new subject in the process

Case 1 refers to situations where only the behavior model of a subject is altered without affecting its interfaces to other subjects. The content, form, and sequence of information exchange remain unchanged. In this case, the changes only affect one actor and do not require changes in the behavior model of other subjects. An example would be extending the behavior model of a subject with an additional optional activity that, in some cases, might be needed to prepare information for an activity already represented in the model.

Case 2 refers to situations where the content, form, and sequence of information exchanges remain unchanged, but responsibilities are shifted from one subject to another. In this case, the affected function state must be incorporated in the behavior model of the target subject. An example would be shifting the responsibility for an already existing activity from one department to another, each of which is represented as a subject in the work model.

Case 3 refers to situations where the sequence of information exchange is altered; however, both content and form remain unchanged. In this case, the subject partnering in the communication needs to adapt its behavior model to fit the new expectations. This might trigger subsequent changes in this subject, which again potentially causes cascaded changes elsewhere in the process. An example without cascaded changes would be altering the sequence of information exchange between the one department and another to optimize for time efficiency in communica-

tion. This causes changes in the targeted department, as it has to be prepared to accept and process information already at an earlier point in time in the process.

Case 4 refers to situations where the information exchange is fundamentally altered in a way that adds or removes communication acts to or from the behavior models of the involved subjects. This necessarily causes changes in the targeted subject's behavior model, as it needs to react on new information or provide information that was not expected before the change. This again potentially causes cascaded changes elsewhere in the process. An example would be the decision of one actor to alter communication policies for decisions within an organization. The original way of communication thus is changed and requires the originally involved actors to change the behavior models of their respective subjects.

Case 5 refers to situations when a new subject is added to the process. This requires specifying the communication interface (i.e., the information exchange) with this new subject as well as its behavior if it is known and relevant to the work process. Adding a new subject might have implications on the behavior of the other involved subjects, as additional information exchange might be required. An example would be the addition of an additional actor for decision-making in a work process for specific cases. In this case, the behavior of the other actors needs to change to communicate with the new actor.

In case a change of a behavioral model of a subject triggers the need for changes in the models of other subjects (i.e., in cases 2–5), those changes do not necessarily need to be made immediately. Models of subjects' behavior are only loosely coupled and are basically executed independently. Necessary changes in a subject's behavior, such as addition or removal of activities or messages, need to be kept track of by the enactment instrument and only need to be considered before execution of that respective subject continues. Considering the example used in case 4, this means that the one actor could change his/her process and continue to describe the altered activities from his/her own perspectives. The need for changed communication interfaces and behavior for the other actors would be logged and can be handled when execution of their respective parts of the process continues. The refinement of the process, however, can only be finished, once all open change requests have been resolved by

either incorporating them in the targeted subject's behavior model or undoing them in the originating subject's behavior model.

Process validation and elaboration through enactment is a means to complete a process description without the need to create comprehensive formal process models by traditional conceptual modeling. Separation of models and model changes along the different involved subjects reduces complexity and allows focusing on a single subject's behavior at a time. Using the execution engine supports simulating complex decision processes by incrementally adding process variants to the model as the simulation continues. Complex models of collaborative work processes emerge in this way without the need to ever translate one's perceptions of a work process to abstract process descriptions. Still, as the model emerges, it permanently maintains a syntactically valid state that allows for further processing, such as live validation of deadlocks, life-locks, or mathematical simulation of capacities.

5.2.3 Tool Support

We here assume 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 walk-throughs, 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 (Hjalmarsson et al. 2015; Herrmann and Loser 2013). The importance of dynamically adapting the level of support depending on the level of experience or expertise of the stakeholders is especially stressed in the study by Hjalmarsson et al. (2015) 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 (Van de Pol et al. 2010). The overall conceptual approach is thus termed “virtual enactment through scaffolding.”

5.2.3.1 Conceptual Considerations

Virtual enactment, as specified here, draws from the concept of facilitated model walk-throughs (Herrmann et al. 2004), which are usually carried out in a co-located group setting, bringing together all involved stakeholders at the same place and at the same time to collaboratively go through the process. A distributed form of model walk-throughs can be imagined (and actually has been considered in our earlier work; Wachholder and Oppl 2012, 2014) but is not subject of the presented approach due to the inevitable loss of communication and negotiation potential that would need to be compensated for by further groupware instruments.

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, that is, 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 observing the progression through the process across all involved process actors, maintaining an overall bird's eye 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 more than a single path through the process can be enacted at a time. Hence, there is no 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 (i.e., actor in a specific role) 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 actors 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 (Oppl 2016).

Elaboration of the process should be possible during enactment, whenever the need arises, without having to start over the enactment process from the beginning. Such a feature avoids losing the context of the current walk-through and further does not distract stakeholders following their current line of thoughts, when 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. Oppl 2016). 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., deadlocks or non-terminating loops). Actors need to be pointed to 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 provided with scaffolds 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.

5.2.3.2 Architecture

Based on the concept and requirements described in the previous 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 the features of the implemented modules.

The virtual enactment platform is implemented in a modular way and accessible to users via a web-based interface. Figure 5.5 gives an overview of 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.

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 the selected 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 S-BPM approach, composed of Subject Interaction Diagrams and Subject Behavior Diagrams, as introduced in Chap. 3.

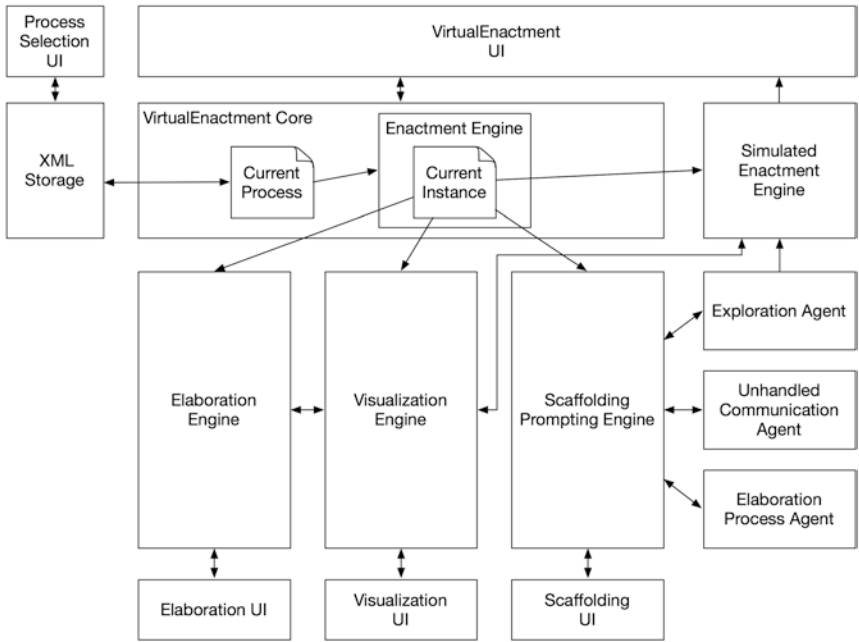


Fig. 5.5 Platform architecture

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 they 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 or 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 later.

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 defining new subjects, their behavior, and their interaction using the same prompting mechanism as deployed for elaboration).

5.2.3.3 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).

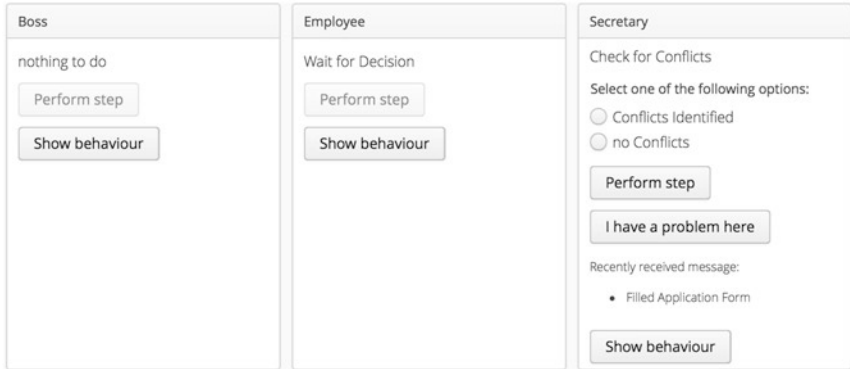


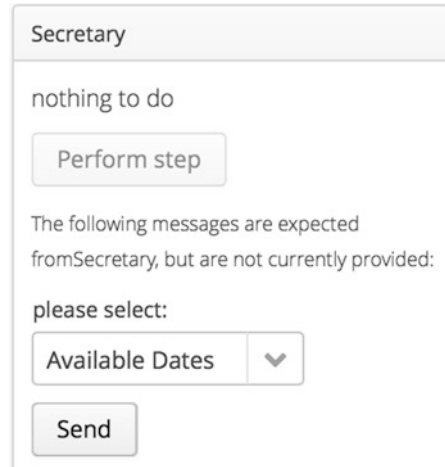
Fig. 5.6 Enactment UI (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

Incoming messages are stored in an input pool, from which they are removed as soon as the respective receive state is triggered.

Figure 5.6 shows an example of the enactment UI. As can be seen, subject *Secretary* currently is in action state with two potential outcomes (cf. Fig. 5.8 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 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 (see later), while the button labeled *Show behavior* triggers the visualization engine to display the respective Subject Behavior Diagram.

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. 5.7.

In turn, subjects might also provide messages to other subjects, who do not react at these messages in their current behavior. Such messages are added to the envisaged recipient's pool of provided messages. 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



Secretary

nothing to do

Perform step

The following messages are expected
fromSecretary, but are not currently provided:

please select:

Available Dates ▼

Send

Fig. 5.7 Expected messages in subject UI (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

incorporated in a subject's behavior by adding send and receive states, respectively.

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.

5.2.3.4 Visualization Engine

To enable users to create a link between the current state of the simulation and the underlying model, visualizations of the model can be displayed at any time during exploration. The visualizations are available in different levels of complexity and from different perspectives on the process (view per actor, overall actor-centric view, overall flow-oriented view), and are augmented with information about the current instance, such as the currently available activities and the path through the process. The visualizations are created dynamically using the GraphViz software suite (Ellson et al. 2001).

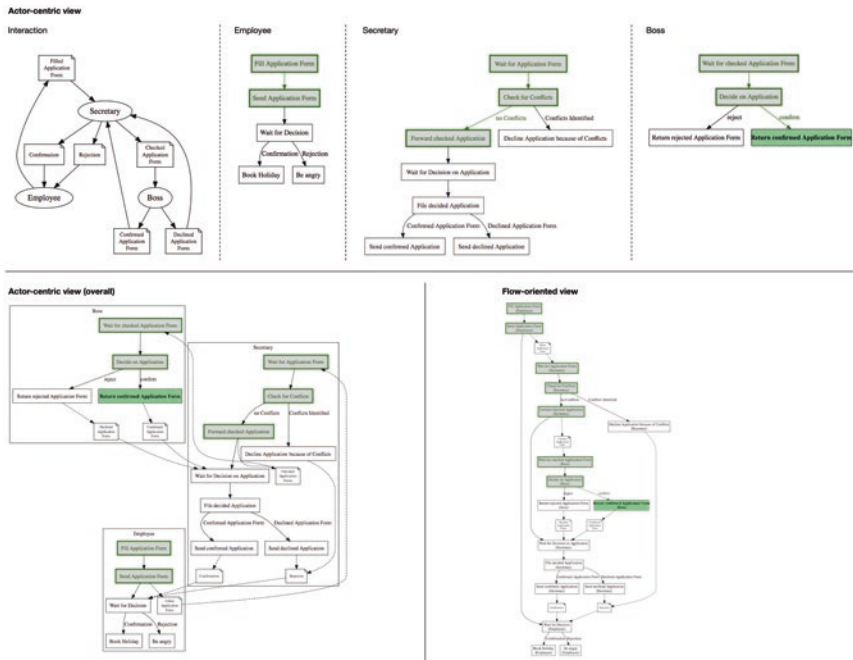


Fig. 5.8 Process visualizations (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

Figure 5.8 shows visualizations for an instance that has been simulated halfway through a sample process. The four models at the top of the figure together form the least complex visualization, where the behavior of each actor is shown as a separate model. The model at the extreme left shows the interaction among the actors. The gray boxes indicate already executed activities, whereas green boxes represent currently available activities. The lower left model in Fig. 5.8 compiles the separate actor models in a single visualization and enriches them with connections representing the exchanged messages. The lower right model removes the actors as the primary structuring dimension for the overall model and, in this way, provides a flow-oriented view on the process. Users can switch between the actor-specific behavior models, the interaction overview and the two overall views at any point in time and, in this way, can focus on different aspects of the model in the course of simulation or reflection.

5.2.3.5 Elaboration Engine

The elaboration engine allows modifying 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 a process in the context of that particular state.

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. Figure 5.9 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.

An example for an interactive prompt is shown in Fig. 5.10. It shows the user interaction for the element labeled *specify new step or select existing one* in the topmost branch in Fig. 5.9. 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.

Wherever necessary, the prompts dynamically adapt to the current state of the process. Figure 5.11 shows an example for this feature. It shows the user interaction for *specify message & recipient* as referred to earlier.

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. 5.9). If users choose to specify a new message to be provided to others (as shown in Fig. 5.11), the list of

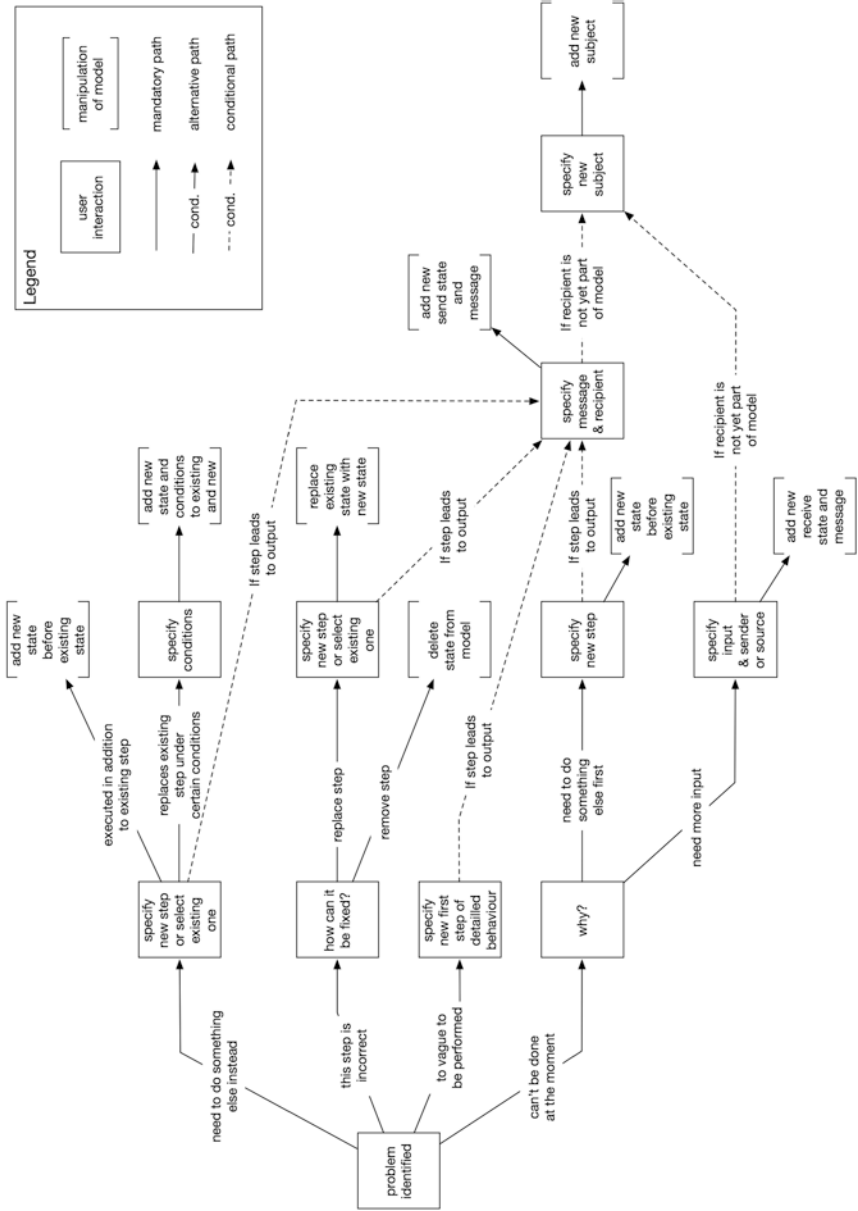


Fig 5.9 Prompting sequence for elaboration

Elaborate on this problem + ×

I need to do something else instead of "Check for Conflicts".

What do you need to do?

☐ This step leads to results I can provide to others.

How does this relate to "Check for Conflicts"? ☐ It replaces "Check for Conflicts" under certain conditions.
☐ It is complementary to "Check for Conflicts", I still need to do "Check for

Fig. 5.10 Example for interactive elaboration prompt (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

Elaborate on this problem + ×

"Check for company wide constraints" leads to results I can provide to others.

There are some expected results, which you currently do not provide: ☐ Available Dates
☒ I can provide other results.

What can you provide to others?

Whom do you provide it to? ☒ Employee
☐ Boss
☐ Somebody else.
☐ I do not know who could be interested

Fig. 5.11 Specification of messages during elaboration (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

potential recipients is dynamically created from the set of subjects currently contained in the process model. The option *Somebody else* consequently would trigger the option path leading to the addition of a new subject. The 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, that is, 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.

5.2.3.6 Simulated Enactment Engine

Playing through instances of complex processes several 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.

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, the states to be executed and the 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. The steps are executed with short delays in between, making it possible for users to follow the simulated enactment process on the UI. Simulated enactment stops at the requested end state and hands back control to the users for further manual enactment of the currently running instance.

5.2.3.7 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 whether a new instance is going to be started. Concrete examples of different scaffolding agents are described later.

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. Figure 5.12 gives an example of the selection of scaffolds displayed in a table-like form.

The slider bar on the left border of the area can be used to adapt the concreteness of scaffolds to be displayed. Depending on the requested level of concreteness, the engine displays either procedural (most concrete), strategic, metacognitive, or conceptual scaffolds (least concrete). Placing the slider at the bottom turns off the display. A scaffolding agent consequently provides groups of scaffolds of different types on a particular issue. For instance, such a group can contain a procedural scaffold and a metacognitive scaffold, omitting strategic and conceptual scaffolds. The engine then displays the most concrete scaffold for the level requested by the users. If users requested strategic scaffolds, the metacognitive scaffold would be displayed.

A further level of user control with respect to displayed scaffolds is the ability to dismiss scaffolds. The engine keeps track of dismissed scaffolds and does not display them as well as 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.

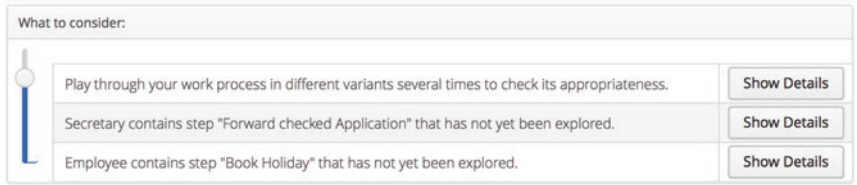


Fig. 5.12 Scaffolding prompts (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

The text-based prompts displayed in the table can be detailed in arbitrary ways. The current implementation—besides the basic single-line prompt-only scaffold—offers a type of scaffolds that can display further information in a pop-up window upon request, and a type of scaffolds that can trigger the simulated enactment engine to take users to that part of the process which the scaffold suggests exploring further (cf. Fig. 5.13).

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.

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 to handle the platform. Consequently, it provides scaffolds, introducing the features of the platform as being distributed in the first few executed instances. While initially users are only asked to explore the process using the execution UI, the agent gradually offers scaffolds introducing the visualization UI and the elaboration UI. In this way, users are introduced to the platform features step by step.

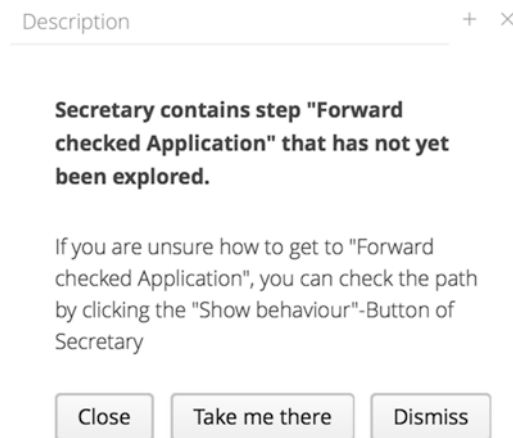


Fig. 5.13 Example for exploration scaffold (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

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

Figure 5.13 shows a strategic scaffold pointing at an unexplored part of the behavior of subject *Secretary* (behavior shown in Fig. 5.8). 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).

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 directs them towards, resolving such modeling issues.

Figure 5.14 shows a metacognitive scaffold for the fact that the subject *Secretary* has expected messages, which is currently not provided by its

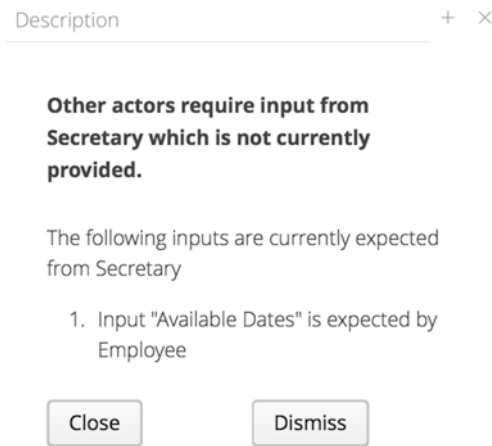


Fig. 5.14 Example for unhandled communication scaffold (released under a Creative Commons Attribution 4.0 International License (CC BY 4.0))

behavior specification. 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 due to user elaboration activities, this agent registers not to be updated per completed instance, but per each completed execution or elaboration step.

5.2.4 Conclusive Summary

The goal of exploring a business process is to understand the depicted information fully and to compare this information to the actual perceived context of the work process, that is, whether the portrayed information matches informal specification as established during the information elicitation phase. Usually, there is a rather strict distinction between the tasks of a system analyst and domain experts in a modeling process (Frederiks and van der Weide 2006), which leads to model alterations not being part of the validation process itself, but rather between validation iterations, as a task that system analysts take over. However, as validation through virtual enactment aims at reducing the need for facilitators and system analysts during the validation phase, this step is considered part of validation. Finally, consolidation activities are required in case different domain experts or stakeholders have opposing views on the depicted information or possess different points of views about the reality the model is an abstraction of.

Reflecting these validation steps against the presented instrument's capabilities and functionalities, the following observations can be made: there are multiple possibilities to explore a business process, utilizing the platform. First, the main intended usage of exploring the process is by enacting it akin to a role-playing game. As business processes in most cases are not comprised of only a single, linear path but exhibit multiple decision points resulting in forking paths which eventually are joined together again, it is usually not possible to explore the complete business process during one instantiation. In addition to process enactment, it is also possible to visualize the process in various ways, such as depicting a Subject Behavior Diagram for each actor, a Subject Interaction Diagram,

or the whole process as a control flow diagram. The main target audience of the virtual enactment instrument include domain experts and stakeholders who are participating in the represented work process and are assumed not to be experienced in modeling notations or visualization methodologies. Accordingly, visualization methodologies should be seen as an enhancement of process enactment and, with regards to the target audience, not as the main means of process exploration.

Although making changes to the business process model is not an explicit part of the validation from the domain experts' point of view, due to factors already mentioned, it is an integral part of the business process validation in conjunction with the instrument. Changes to the underlying model can only be made while the process is currently enacted. Although constraining the freedom to change the model, it enables users to alter the model directly where the need for changes arises, thereby reducing the cognitive load (Forster et al. 2013) of having to remember those activities that are not reflecting reality in the desired way. Additionally, the instrument does not require its users to possess prior knowledge of modeling concepts, modeling notations, or to completely understand the visualizations of the model. Model alterations are conducted by following a series of elaboration prompts, using natural language, ultimately leading to the anticipated changes to the model.

The instrument does not explicitly implement means to support consolidation activities. However, this does not necessarily mean that it is unable to support the validation of business processes for that reason, since the platform is currently implemented in a way that requires process participants, that is, actors, to be in the same room, or rather in front of the same screen. Therefore, it can be argued that different point of views or varying perspectives which require consolidation activities can be resolved via means of face-to-face discussions.

5.3 S-BPM-Driven Execution of Actor-Centric Work Processes

Subject-oriented Business Process Management (S-BPM) (Fleischmann et al. 2012) explicitly considers the role of actors during process design and execution. The primary elements of structuring are subjects that sep-

arate a process along who performs work in the process. How to deal with knowledge, however, is not explicitly taken into account in this approach.

Knowledge-intensive business environments (Dalmaris et al. 2007) claim for support measures that allow for actor-aware and agile process management (Bruno et al. 2011). Agility allows for overcoming contingencies during work on an ad-hoc basis (Minor et al. 2008), and actor awareness enables individualized workflow execution (Prilla and Nolte 2012). Individualized workflow execution does not restrict actors in how they perform their work while still maintaining reliable interfaces to others and pursuing the overall work goals. It furthermore enables situation-specific IT support that is tailored to both, the current work aim and the person executing the work (Monsalve et al. 2010).

The operative aspects of S-BPM are basically specified by a set of actors carrying out different activities in a set of activity bundles. In contrast to other BPM approaches (for a comprehensive overview cf. Weske 2010), the BPM activity bundles are not specified to be carried out in any particular order (whereas most other approaches follow a cyclic approach that is implemented by iteratively stepping through the different phases of business process management).

S-BPM specifies a set of four roles that are relevant when implementing the activity bundles (Fleischmann et al. 2012). They can be linked to the concepts developed by Firestone and McElroy (2003) in the Knowledge Lifecycle (KLC) Framework as described in Chap. 1, thus intertwining work modeling activities with knowledge management processes:

- *Actors* are people that are actively involved in the work process. In S-BPM, they are the source of process knowledge and the ones who put this knowledge to practice again. In relation to the KLC, actors are the primary entities in the business processing environment and thus are the main triggers of learning processes.
- *Governors* are responsible for development, implementation, and monitoring of processes from an organizational perspective. They define the general condition under which a process is implemented and can be altered. While their role is limited in the immediate activities of the KLC, they determine the fundamentals under which processes can be improved in an actor-centric way.

- *Experts* are providers of knowledge that none of the involved actors have in a specific situation. They support activities in all activity bundles of S-BPM with their expertise. Their role in the KLC is also spread across support during business processing and knowledge processing.
- *Facilitators* aid organizational development and provide support during activities, leading to process change. They are not directly involved in operative work activities. In the context of the KLC, their role is mainly important in the knowledge processing environment.

All four roles work together to implement process improvement and adaptation to the current organizational situation. The activities leading to this change are structured into seven activity bundles (cf. Fig. 5.15). As mentioned earlier, these bundles can be executed in arbitrary sequence depending on the current situation in which process change is triggered. Not only sequence but also the actual activities carried out during these bundles are not fully pre-specified and again depend on the situation in which an activity bundle is triggered (Fleischmann et al. 2012). We can implement an instance of the S-BPM activity bundle that is specifically

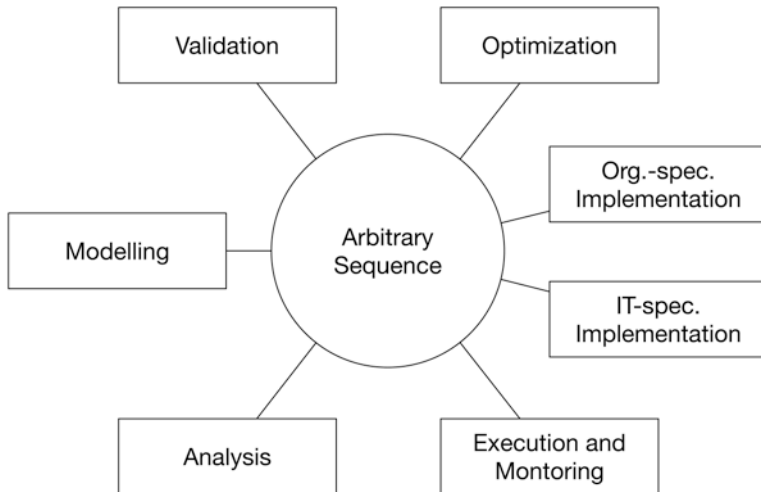


Fig. 5.15 The S-BPM activity bundle (adapted from Fleischmann et al. 2012)

tailored to support the KLC in both, possible sequences and methodological support for activity bundle implementation.

- *Analysis* covers activities concerned with examining a process and the conditions under which it is executed. It collects and structures information about a process and thus provides input of a number of other bundles that are concerned with process change. This bundle might be triggered externally to set up a new process, but also from monitoring during process execution, especially when a mismatch between expected and actual outcome occurs.
- *Modeling* refers to all activities that deal with conceptualizing a process by abstracting it from its real-world execution environment and describing it as a generic phenomenon that can be reproduced given certain conditions. Process models cover different aspects of work (depending on the chosen modeling approach) and conceptualize them in form of a graph. In general, at least information about the activities to be carried out (what?), their sequence (when?), the actors carrying out the activities (who?), and which resources are required (with what?) is represented. Specifically, the who-dimension is used in S-BPM as the primary dimension for structuring the models, which makes it especially suitable for representing processes from an actor's perspective. Actors can focus on their own activities and their interaction with others without the need to have an overall view on the whole process.
- *Validation* is the activity bundle concerned with testing the effectiveness of a process, that is, if its implementation leads to the desired outcomes. Validation is not necessarily carried out in the real work situation but can also be performed in artificial, simulated work situations that allow for testing different relevant aspects of the process (depending on which parameters are considered in setting up the artificial situation).
- *Optimization* refers to the activities that improve the efficiency of processes. Originally described to focus on an economic approach to efficiency, optimization in this context refers to activities that improve the implementation of an already existing process without questioning it. This activity bundle is mainly triggered if parameters of the organizational situation in which the process is carried out in change and the expected outcome cannot be reached anymore.

- *Organization-specific implementation* deals with all activities that are necessary to implement a new or changed process within the organization, that is, rolling out information about the changes or changing organizational structures and responsibilities, when necessary.
- *IT-specific implementation* covers activities that are concerned with implementing a new or changed process in an IT-supported environment, for example, using a workflow management engine or groupware systems.
- *Execution and monitoring* finally cover all activities that are carried out during a process conducted by actors in a specific organizational situation. Monitoring is especially relevant for identifying deviations from expected process outcomes and triggering activity bundles to compensate these deviations.

The activity bundles of S-BPM can be linked to the KLC at several points. Integrating S-BPM with the KLC leads to an instance of the KLC that is methodologically augmented to handle business process knowledge. The activity bundles of S-BPM will be put into relationship with the building blocks of the KLC in Fig. 5.16, which provides an overview of the integrated framework.

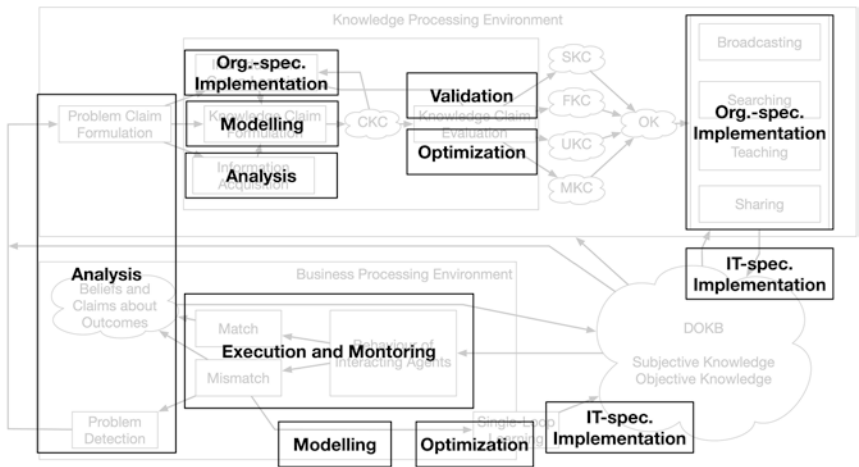


Fig. 5.16 Integration of the KLC with S-BPM activity bundles

5.3.1 S-BPM Activity Bundles in the Business Processing Environment

Execution and Monitoring can be located at the core of the business processing environment in the KLC. It covers actual work being performed by actors (termed “behavior of interacting agents” in the KLC) and the identification of deviations between expected and actual outcome of work (which refers to the “monitoring”-part of this activity bundle). It is important to note that “behavior of interacting agents” does not necessarily refer to the execution of a complete work process (i.e., working until the aim of the work should basically have been reached). The identification of mismatches is an ongoing, parallel activity that can trigger compensation activities, whenever contingencies or unclear situations arise.

The activities during execution are selected and motivated by the contents of the distributed organizational knowledge base (DOKB), which can be codified in IT-support systems and thus be propagated to the business process environment (as, for example, is the case if workflow management systems are used). In this case, the behavior of the actors is directed and coordinated by IT system. In a less IT-supported system, information about how to perform work under particular circumstances may be codified in documents and be provided to the actors for reference. Finally, the DOKB also covers personal experience and expertise of all members of the organization, which also can drive how work is carried out. In real-world settings, a combination of any of these information types will be encountered. The proposed approach especially focuses on settings, where personal experience and expertise plays an important role (i.e., in knowledge-intense processes or expert-organizations, respectively) in an interplay with one of the first two DOKB-sourced drivers of work.

Monitoring covers the identification of matches and mismatches between expected and actual outcomes. In S-BPM, monitoring is originally limited to technical measures to collect performance indicators (e.g., from a workflow engine) and mathematically derive deviations of performance figures from predefined reference values. In combination with the KLC, this understanding of monitoring has to be extended. People involved in the work process can identify mismatches of outcome,

contingencies, or unclear situations based on their knowledge and thus are also triggers for compensation activities. In terms of S-BPM roles, not only *actors* can trigger this behavior from directly within the work process but also *governors* might be able to identify problems or mismatches from their ongoing monitoring activities.

Independently of how the problematic situation or outcome deviation has been identified, the choice of whether to compensate directly (entering the single-loop learning cycle) or trigger a more fundamental examination (may lead to double-loop learning) is a decision to be made by humans (made by *actors* themselves or *governors*). In the case of actors identifying a problem and compensating it directly within the work process, the whole sequence of problem identification, deciding what to do, and performing the compensation activities is not necessarily a conscious process. Still, altering one's behavior due to perceptions of the environment that deviate from what was expected alters how an individual will react in future situations and thus is considered learning (cf. Chap. 1 for a more elaborate argumentation on this claim). If the problem is consciously recognized, actors or governors might still decide not to question the way in which work is performed fundamentally, but to immediately compensate the problems that have occurred (actually, this will be the far more common case).

If immediate compensation is chosen, that is the single-loop learning cycle of the KLC is followed, the business processing environment is not left. The KLC here does not give any clues on which activities are happening in this case. When integrating the KLC with the S-BPM activity bundles, there are several candidates for activities that can be performed in this case. They should, however, be considered optional, as their application depends on the situation the problem was triggered in and the organizational context in which the work was performed (e.g., it makes a difference whether the problem occurred in work process which has been codified in a workflow engine or solely was driven by personal expertise without any external guidance).

Modeling the alternative steps that were taken to compensate the problem can be a first step to persist in learning about the specific situation and potentially make it available to other people in the organizations. Modeling here does not necessarily refer to building a formally

correct and complete business process model but to any form of conceptual externalization of which activities were set in reaction to the particular situation that occurred.

Optimization is the core step of the single-loop learning sequence and covers the activities that adjust aspects of the work process to fit the given situation. This goes beyond the original S-BPM-understanding of this activity bundle that mainly covers measures to increase process efficiency. If a codified model of the work process exists, either being already available in the DOKB or being created on the fly during an instance of the modeling activity bundle, optimization can be considered tinkering the process parameters, such as which and how many people are involved in a certain role in the work process or altering execution and communication patterns to compensate the problems that have occurred. Again, optimization might also be an “invisible” activity, being performed by actors unconsciously.

Finally, if IT-support has been involved in the work process, the altered way of performing work might require adaptation in the IT-systems, which is covered by the S-BPM activity bundle “IT-specific implementation.” This covers changes to workflow definitions (e.g., introducing alternative activities and changing sequences) but also configurations of groupware systems, such as granting additional actors access to communication facilities in which the respective work process is coordinated.

Whatever compensation activities have been taken and whether they lead to externalized, codified results or solely new behavioral patterns of the involved actors, the changes become part of the DOKB and influence future executions of the work process in similar situations.

5.3.2 S-BPM Activity Bundles in the Knowledge Processing Environment

The double-loop learning cycle that involves activities in the knowledge processing environment is triggered either by a problem that cannot be resolved immediately by compensation activities in the business processing environment, or deliberately by actors or governors to revise a work process, that regularly causes problems that can be compensated but

cause overhead and hamper effective and/or efficient execution of work. Triggering the knowledge processing environment in the latter case should be considered an asynchronous activity that does not prevent further instances of the particular work process to be executed in the business processing environment until the double-loop learning cycle is completed.

The block “problem detection” in the KLC refers to identifying problems in the DOKB that lead to the mismatch in outcomes or cause the problems that have been observed. Using the S-BPM activity bundles as a frame of reference, this block already belongs to “Analysis” and leads the way towards leaving the business processing environment and entering the knowledge processing environment. In the course of the analysis activities, the problem in the DOKB is not only identified but being codified in a “problem claim.” A problem claim describes which element in the DOKB is likely to cause the problem, how the problem manifests in the actual execution of work, and under which situation (i.e., conditions in the work environment) it occurs. An element in the DOKB can be anything that bears or codifies knowledge, that is, a set of actors, documents, and workflow definitions.

Based upon the problem claim, the people to be initially involved in the resolution of the problem (referred to as “knowledge production” in the KLC) are identified. This activity and all remaining activities in the knowledge processing environment are driven by a person taking the S-BPM role of a *facilitator*. This person might have been involved in the original work process as an *actor* or a *governor* but can also be a dedicated organizational role or be recruited from outside the organization. Identifying the initially involved actors of the knowledge production process also brings in the people referred to as *experts* in S-BPM, that is, people that have not necessarily been involved in the work process that has triggered the double-loop learning process, but who have expertise that is expected to be relevant for the resolution of the problem.

Activities in the knowledge production process of the KLC are highly iterative and also involve several S-BPM activity bundles. The main goal of these activities is to produce a codified knowledge claim. A knowledge claim, in contrast to the problem claim, describes how the occurrence of the problem can be avoided and eventually has to be integrated in the

DOKB to guide future behavior in similar situations occurring in the business processing environment. The knowledge claim has to be codified to allow for evaluation and distribution across the organization.

The evolution of a knowledge claim is an iterative process involving the KLC activities “information acquisition”, “knowledge claim formulation”, “individual and group learning”, and eventually “knowledge claim evaluation”. In terms of S-BPM activity bundles, this potentially involves all bundles except IT-specific implementation. In the following, the first three KLC activities concerned with the formulation of the knowledge claim will be put into relationship to S-BPM. In a second step, knowledge claim evaluation will be brought together with the S-BPM activities.

The codified knowledge claim is produced in an interplay of explicitly formulating the knowledge claim, developing new ideas as an individual, and transferring knowledge within the group of involved people as well as performing further research to gain more information necessary to develop the knowledge claim.

Knowledge claim formulation is linked with the S-BPM activity bundle of modeling. As mentioned, modeling does not necessarily refer to producing business process models that strictly adhere to the syntax and semantics or to a specific language but should describe how work and interaction are performed in a given situation. Having said that, it might still be necessary to produce formally correct models to allow for simulation, validation, and IT-specific implementation of the model—but that is not a requirement in the early stages of knowledge production. Modeling should be considered a group activity here, as the codified knowledge claim evolves through a cooperative process involving all actors of the knowledge production process (Nolte and Prilla 2012). Individual contributions still can be externalized separately and eventually be brought together in a group process.

Information acquisition instantiates a second iteration through the S-BPM analysis activity bundle. After the formulation of the problem claim or in the course of knowledge claim formulation, the need for further information might become evident. Acquisition activities can include identification of knowledge that led to successful work in similar situations, the involvement of further *experts*, or research activities in

resources stemming from outside the organization (such as scientific or professional literature and case studies).

Individual and group learning finally involve activities to reflect and revise one's own and the whole group's assumptions of how the problem can be prevented to not occur again. This block is not fully covered by any of the S-BPM activity bundles, as they omit the learning perspective on business process management. The activity bundle termed "organization-specific implementation," however, might be triggered in the course of individual and group learning, as the learning processes inherently change how the group of people involved in the knowledge production environment interact. This directly affects future activities in the business processing environment, as, following the S-BPM paradigm, the actors executing work processes are also part of the group performing knowledge processing activities. As long as the actors of the work processes currently being processed are involved, individual and group learning thus will directly contribute to organization-specific implementation of the work process.

A codified knowledge claim that appears to be appropriate to at least some of the involved people participating in the knowledge production process is evaluated in the next step. Knowledge claim evaluation covers all activities that can be performed to justify that the new knowledge claim will appropriately solve or avoid the problematic situation in the business processing environment. This is where S-BPM can contribute most to the activities in the knowledge processing environment—both the activity bundle "Validation" and "Optimization" can contribute to this step in the KLC. Validation refers to activities that check the effectiveness of a process before it is put to practice. Optimization, though not being relevant here as a whole, includes activities to evaluate efficiency of a process and thus assesses the quality of a knowledge claim before putting it to practice. Activities include acting out the process and interacting with the other roles in the process as if it were executed in a real-world setting or simulating process execution by using statistical models. In both cases, ineffective, inefficient, or simply malfunctioning parts of knowledge claims can be identified. Simulation and IT-supported validation are two of the cases that were referred to earlier to require models adhering to formal syntactical and semantic rules. Modeling activities

thus have to include means to guide the involved people to appropriately represent their models.

If evaluation activities identify shortcomings in or inappropriateness of the knowledge claim, this again triggers learning activities which, in turn, lead to revised knowledge claims. If evaluation is finished successfully or further iterations are not considered reasonable, knowledge production ends. The resulting knowledge claim is considered a “surviving knowledge claim” if evaluation was successful. If the evaluation still showed that the problem is not been resolved or evaluation did not lead to unambiguous results and further iterations were not made, the knowledge claim is considered “falsified” or “undecided”, respectively. Still, both of the latter cases are valid outcomes of the knowledge production process as they at least augment the DOKB in terms of solutions that are likely to not work (and thus do not need to be tried in real-world occurrences of the problem).

Together with information about how the knowledge claim has been produced (e.g., who was involved, which information was built upon, how many and which revisions have been made), the knowledge claim needs to be integrated in the DOKB. Again, this step is more detailed in the KLC than it is in the S-BPM activity bundles. Cross-leveling a knowledge claim can be performed by activities like sharing, teaching, searching, and broadcasting (as described in the KLC). All these activities alter how the organization will in future react to the occurrence of the problem that the knowledge claim is intended to solve. They are thus part of what S-BPM refers to as organization-specific implementation. Knowledge claims might also involve adaptations of an organization’s IT-infrastructure (e.g., workflow descriptions, as mentioned earlier). While not an explicit part of the KLC knowledge integration activities, such IT-specific implementations (as referred to in S-BPM) are also part of the activities that lead to integration of the knowledge claim in the DOKB.

The (re-)integration of a knowledge claim in the DOKB ends the double-loop learning process and closes the knowledge lifecycle. S-BPM activity bundles can be found in all steps of the KLC and augment various aspects of it with a more concrete approach on how to implement the steps. In turn, several steps of the KLC, especially those directly concerned

with learning and knowledge transfer, offer a more detailed concept of how to implement those activities than S-BPM does. S-BPM and KLC thus augment each other on a conceptual level and offer a starting point to populate their building blocks with instruments that aid their implementation. Both, the KLC steps and the S-BPM activity bundles, however, do not leave the organizational level when describing process development and knowledge development and knowledge application. To implement the actor-centric approach outlined by S-BPM, a conceptual bridge between the super-individual phenomena described here and actually applicable instruments on an individual or group level needs to be constructed. Articulation Work and Mental Model Theory are candidates to provide the foundations for this bridge and will be reviewed in the following section before an attempt is made to put them into the context of the KLC.

5.3.3 Tool Support

The execution of subject-oriented representation schemes can be supported by an appropriate workflow system (Krenn and Stary 2016). In the following example, we show how the workflow system is used to execute an application for vacation using a generic communication scheme.

Figure 5.17 shows a generic subject-oriented specification scheme with three involved parties. It fits to the holiday application process, as the three subjects are employee (Subject 1), HR department (Subject 2), and manager (Subject 3). Each of the parties exchange messages with another party.

Each subject starting message exchange is marked with a small white triangle (Subject 1).

Each subject can send messages with the name Message to any other subject any time. Figure 5.18 shows the behavior of the subject with the name Subject 1. Since Subject 1 is the subject who starts a process, its start state is the state select. The start state is marked with a thick frame. The state “start” and the transitions to the state select will be never executed in the start subject. This state is the start state in all the other

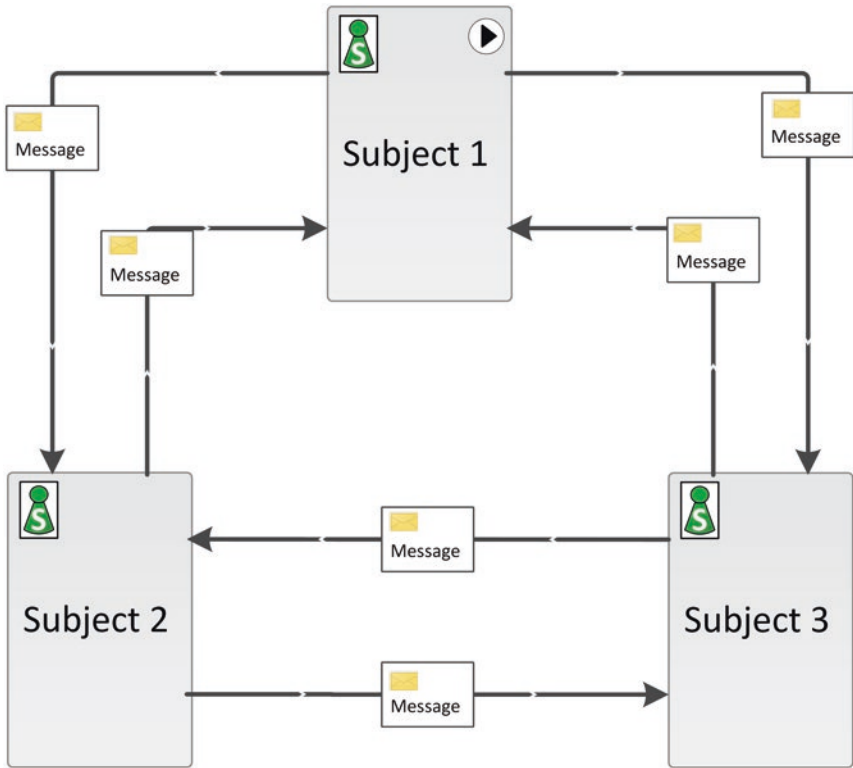


Fig. 5.17 Subject-oriented representation schema for three-party process

subjects. All the other subjects are waiting for a message from all the other subjects.

In this way all subjects that are not start subjects have to receive at least one message before they can start to send messages. The start subject sends a message to any other subject. The receiving subject can now reach the state select. In that state, any subject can decide upon its next action without restriction. A subject which is in state select can send a message to other subjects which are still in the state start. Now these subjects can also reach the select state and can send messages. Finally, all subjects are in the state select and can communicate when addressed.

In the “select” state, the start subject decides whether it wants to send or to receive a message. In order to start a workflow, it does not make sense to receive a message because the other subjects are waiting for mes-

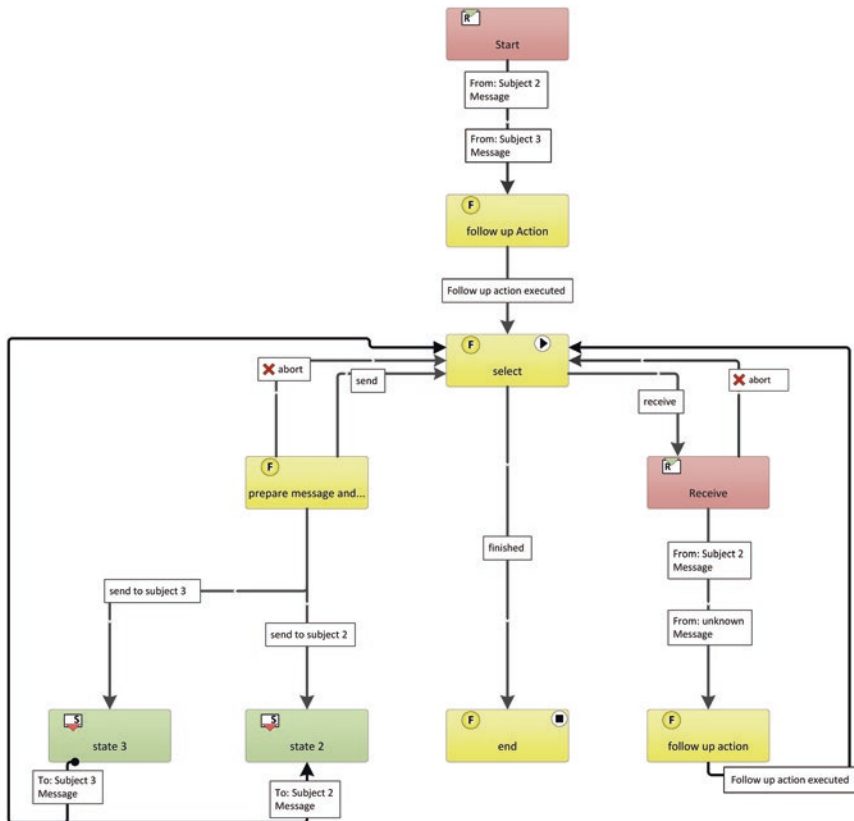


Fig. 5.18 Generic behavior of the start subject "Subject 1"

sages. All the other subjects are in the state start which is a receive state (cf. Fig. 5.19). This means the start subject will start with sending messages. Now the message exchange can begin. In the select state, a subject decides to use the send transition. In the state "prepare message and select address," the subject fills out the business object that is transmitted by the message "message." After that, a subject decides to which subject the message with the business object as content will be sent.

In the select state, a subject can also decide whether it wants to receive a message. If a message from the expected subject is available, the message can be accepted, and a follow-up action can be executed. It is not specified what the follow-up action is. This is like receiving an e-mail. The

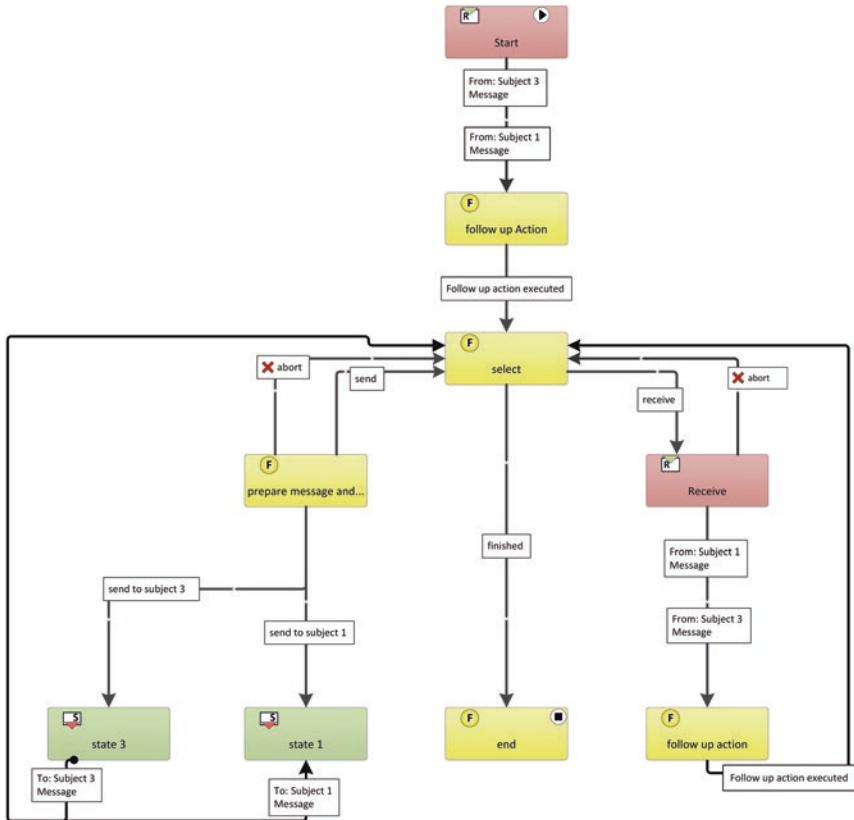


Fig. 5.19 Generic behavior of “Subject 2”

receiver can interpret the content of an e-mail and knows what the corresponding follow-up action is. The abort transitions back to the select state enable to step back in case a subject has made the wrong choice.

With the message “Message,” a corresponding business object is sent. The structure of this business object corresponds to the structure of a mail with some extensions like keyword and signature. Figure 5.20 shows the specification of the business object message in an XSD notation.

Whenever a message “Message” is sent, such a business object is sent. The values for the components of the business message object correspond to the content of a traditional mail.

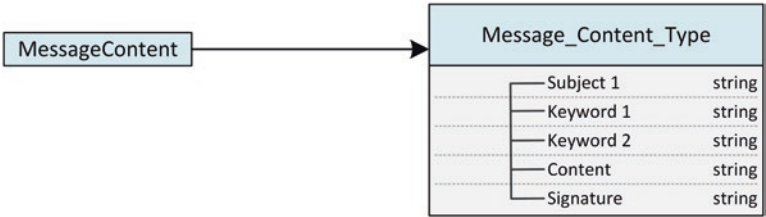


Fig. 5.20 Generic structure of the business object “Mail”

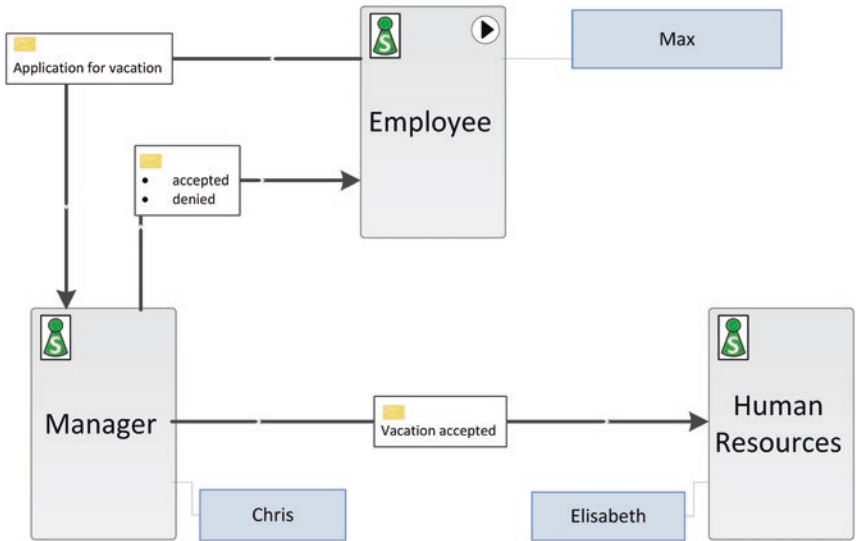


Fig. 5.21 Instantiating a process scheme

For the specification of an actual workflow, the various subjects of a process must be assigned to existing roles and persons or agents. The example shown in Fig. 5.21 demonstrates such an assignment to the three-party process scheme.

The workflow support system is configured in a way, that the actors Max, Helge, and Josi can be assigned to subject “employee.” Since these actors are assigned to the start subject, all of them can start the process. For instance, Max creates a process instance and is then guided through the process. He is asked by the workflow system about the transition he

wants to follow. He knows that he has to fill out the business message form with the corresponding data and that form has to be sent to Chris. Chris who is assigned to the subject “manager” can accept that message and can send a message Accepted or Denied back to subject employee—the message is received by Max because he is assigned to subject employee. Max receives the message because in his environment or context, the process is started. If another person assigned to subject employee starts a process, this process instance is executed in his or her environment. Elisabeth from HR (Human Resource department) receives already accepted holiday requests and processes them accordingly.

In the course of process execution, support for disseminating knowledge that is represented in the distributed organizational knowledge base (DOKB) as conceptualized in the KLC is also required for effective knowledge use. As described earlier, the DOKB does not solely contain technically represented, automatically executable knowledge claims. It rather conceptually includes all knowledge of the members of an organization, their experiences and expertise, as well as all individual and organizational procedures and facts that have been codified in IT-systems, workflow specification, or other organizational descriptions.

In the context of the KLC, dissemination of content of the DOKB is necessary in four different cases, namely

- during operative work activities in the business processing environment
- in the course of identifying and formulating a problem claim to trigger activities in the knowledge processing environment
- during knowledge production
- during knowledge integration

During operative work activities, the actors need to be able to access the DOKB in order to decide on how to react on the currently perceived situation in the work environment. In terms of accessing the explicitly codified part of the DOKB, this can be realized by providing in situ scaffolding, that is, providing suggestions on how to continue to actors based either on their own previous task implementations or on other’s task implementations (“what can I do now/next?”). Alternatively, actors also

need to be able to assess whom to approach, if they cannot decide on how to proceed (“whom can I ask?”).

In the course of formulating a problem claim, access to documentation of previously executed processes of the same process class is necessary. This allows identifying potential sources of the problematic situation and reflecting upon the different variants on work execution that might provide deeper insights in how the addressed problematic situation has developed.

Activities in knowledge production again require access to historic process execution information and transactive knowledge of which information has been codified, under which circumstances, and by whom. Sharing of instance knowledge between the involved actors is required to facilitate alignment of meaning as well as for development and evaluation of the knowledge claim.

Knowledge integration aims at disseminating a new knowledge claim to the organization, eventually making it part of the DOKB. For all activities carried out in this step of the KLC, it is necessary to put the new knowledge claim into the context of the current state of knowledge of the targeted actors. Knowledge dissemination thus not only has to be tailored to the specific situation during a work process but also has to allow actors to put the new knowledge claim in the context of their own mental model during dedicated learning activities in knowledge integration.

Summarizing, support for knowledge dissemination should support the following scenarios for dissemination knowledge about work processes:

- In situ scaffolding during operative work processes (what can I do now/next)
 - based on previous own task implementations
 - based on others’ task implementations
- Learning from other actors who previously took the same role in a task bundle or are experts in the area during operative work processes or in knowledge integration (whom can I ask)

- Reflection on previous own and others' task implementations as a part of problem claim identification and knowledge production (what have we done)
 - individually
 - as a group
- Aligning with other actors who take other roles in a task bundle during knowledge integration (how can we work)

These requirements can be methodologically and technically approached using instruments for self-directed learning. Based on the learning support platform (Auinger and Stary 2005), concepts have been developed to extend learning support to organizational learning settings (Neubauer et al. 2011, 2013). We will re-address these concepts in the digital work design framework developed in Chap. 6.

5.4 Synthesis

Table 5.1 gives a structured overview of the presented techniques of this chapter. It structures each approach according to its

- focus revealing its objective
- essential support features
- means of representation, in order to validate and process documented work knowledge
- procedure to follow for putting process knowledge to work practice

Validation and enactment of work models can be facilitated by scaffolds and subject-oriented processing. Role-specific behavior representations on various levels of detail play a crucial role in stakeholder-driven reflection of the (future) organization of work tasks. Scaffolds open up for additional articulation while execution enables interactive experience of models in terms of actual process support of individual work practices in the selected business process (Table 5.2).

Table 5.1 Processing work models for validation and enactment

	Virtual enactment through scaffolding	S-BPM-based validation and execution
Focus	Sharing simulated process-model walk-throughs in a role-conform way	Building interactive process experience enabling role-specific semantic behavior checks for a specific work process
Essential support feature	Adjusted scaffolds guiding validation	Software engine supporting the execution of interacting behavior encapsulations in a role-specific way with a dual view on models and user interactions
Means of representation	Workflow DiagramInteraction Relations	Subject Interaction Diagram (SID)Subject Behavior Diagram (SBD)
Validation and implementation procedure	<ol style="list-style-type: none"> 1. Scaffold preparation 2. Participatory simulation preparation 3. Detailing specifications 4. Collective reflection 	<ol style="list-style-type: none"> 1. Provision of Subject Interaction Diagram 2. Provision of Subject Behavior Diagrams corresponding to 1 3. Preparation of execution environment 4. Execution of behavior diagrams 5. Reflection of interactive process experience(s)

From a procedural perspective, scaffolding for virtual enactment comprises several phases:

1. It involves preparation of the setting, actors, and instruments, comprising (i) determining the type of scaffolds and scaffolds that guide the implementation activities, (ii) instrument support including digital media support as execution environment, (iii) actors willing to take responsibility for validating role-specific behavior and rethinking, (iv) a facilitator to guide the validation procedure and use of enactment environment.

Table 5.2 Elicitation requirements and scaffolding-based validation and virtual enactment

Elicitation requirement	Scaffolded virtual enactment
Awareness on role(s) and their management	Walk-throughs in the course of validation and alignment are performed along role-specific behavior representations including their interactions with other roles. Depending on the type of scaffold, further articulation of work knowledge could be required and it needs to be aligned among role carriers. As such, the approach can be considered as active role management—determining the various role behaviors develop over time and can be re-arranged on the fly using the corresponding digital tools.
Situation awareness	Validation is framed by situation information as defined at the core of role behaviors. The participatory and multifaceted scaffolding approach helps creating situation awareness in the participating group of stakeholders.
Conceptual understanding of complex systems	Complexity of systems can be addressed by the variety of scaffolding approaches that can be applied for the role-specific walk-throughs. Rather than prescribing linear (for the roles) and networked (by interaction between roles) validation and enactment, work knowledge can be represented according to the capacities of the participating stakeholders. Complex system specification can develop step by step. A further facilitator is the switching between views, referring to actors or the flow of work in the course of virtual enactment.
Creating a reflective practice for situations-to-be	Future actor behavior can be compared with originally articulated and previously aligned representations via the models that are created through scaffolding and virtual enactment support. Again, the participating stakeholders can switch between views, either referring to actors or the flow of work.
Focusing while utilizing multiple perspectives	On the one hand, the different types of scaffolds enable various perspectives on work processes with respect to the completeness or prescriptiveness of specifications. On the other hand, the virtual enactment support allows taking various views on processes, namely, view per actor, overall actor-centric view, and overall flow-oriented view, and is augmented with information about the current instance, such as the currently available activities and the path through the process.

(continued)

Table 5.2 (continued)

Elicitation requirement	Scaffolded virtual enactment
Articulating intangible assets	Based on the view per actor, overall actor-centric view, and overall flow-oriented view, and the information about the current instance, visualizations are created dynamically. Hence, becoming aware of intangible assets relevant for the work process at hand could easily be embodied in terms of encoded process elements.
Engage in alignment for collective intelligence	Although the focus is on role-specific behavior, the overall flow-perspective allows all stakeholders to engage in experiencing models by walk-throughs and share the documented knowledge, guided by the prompting mechanism when probing novel organizational behavior.

2. Situation-sensitive articulation is supported through the scaffolds and the enacting environment when aiming to implement specifications of work processes.
3. Facilitation is required (i) to set the stage involving stakeholders as role carriers, (ii) to ensure the usefulness and effectiveness of the selected scaffolds, and (iii) to support the use of the virtual enactment tool.
4. Representational alignment might need to be facilitated when the participants aim to consolidate their findings into a shared representation in the course of enactment.
5. Organizational implementation needs to be documented when the participants validate how work processes could become part of future workplace designs.

We now discuss the requirements with respect to subject-oriented validation and execution based on S-BPM (Table 5.3).

From a procedural perspective, S-BPM-based validation and execution needs to take into account the following sets of activities:

1. The preparation of validation and execution requires (i) determining the models within the scope of implementation, that is, some business process, (ii) providing the models and the probing actors as role carriers, (iii) configuring the tool for validation and execution of the role-specific models, (iv) providing a facilitator to effectuate the procedure including articulation of additional work knowledge when probing.

Table 5.3 Elicitation requirements and S-BPM-based validation and execution

Elicitation requirement	S-BPM-based execution
Awareness on role(s) and their management	Roles and role-specific interactions are constitutive elements of subject-oriented representation that lay ground for validation and implementation. Automated execution of Subject Behavior Diagrams helps to experience role-specific behaviors, and thus manage roles based on their function flows and interaction patterns.
Situation awareness	The selection of roles and thus subjects depends on the situation to be addressed in the course of articulation, alignment, and implementation. Consequently, a situation is grasped and modeled implicitly through the resulting interaction pattern between roles and their behavior specifications.
Conceptual understanding of complex systems	A system can be validated as being composed of an arbitrary set of entities. The ultimate criterion is whether the constellation of subjects encapsulates intended work behavior. Validation can be achieved by specifying subject behaviors and their interactions, and by executing these representations by choreographic workflow engines. According to S-BPM methodological approach, several facilitating organizational roles can be identified for validation and execution. They are either governing or acting throughout implementation, aiming to build up digital work experiences based on validated process models.
Creating a reflective practice for situations-to-be	The baseline in S-BPM can either be a situation as-it-is, or a situation to-be, or even a mixture of existing and envisioned patterns of work. It depends on the purpose of validating knowledge and the status of implementing models.
Focusing while utilizing multiple perspectives	Each subject pursues a specific role perspective in the course of validation and execution. The overall picture becomes visible through interaction between the subjects. When executing the models by digital means, the overall process flow, as well as the individual models of the role-specific behavior, is available. Hence, for probing and interactive process experience, a dual view (design and runtime) can be provided. In addition, the procedure of validation and execution is supervised by additional organizational perspectives, in particular through the governor and facilitator role.

(continued)

Table 5.3 (continued)

Elicitation requirement	S-BPM-based execution
Articulating intangible assets	Subject-oriented validation is based on explicit work models, in particular, it allows to explore how work task are accomplished when collaborating in a certain role with other roles. Whenever a participating stakeholder becomes aware of work knowledge not externalized so far, it can be articulated in the course of model validation and represented to become part of the subject representations that can be executed.
Engage in alignment for collective intelligence	A subject-oriented business process consists of collaborating subjects. For validation and implementation, all involved stakeholders can experience how models could be implemented in work practice. Hence, it is advised from a project management perspective to involve the actual takers of each role to check whether the models provide a balanced perspective on individual mental models of work. Otherwise, the impact of implemented work processes on the collective intelligence could only be evaluated after probing the models.

2. Situation-sensitive validation and execution features are modeling and tool functions referring to the notation (e.g., editor) and execution support (e.g., bootstrapping through subject-wise execution), as the subject constellation refers to the situation to be targeted. It is captured through a functional and interactional perspective.
3. *Situation-as-is versus situations-to-be*: Both can be addressed by models when validating and executing them as being constructed. Hence, also the transformation from as-it-is to as-it-could-be can be experienced interactively, which is particularly helpful when stakeholders start revisiting existing work patterns, role labels, and work assignments, and try to generate novel structures of work. Our practical experiences show a strong preference of stakeholders of this middle-out approach.
4. *Representational validation*: The specification of work processes is complete with respect to the information required for executing models without further transformation. Hence, stakeholders are enabled to change patterns of behavior back and forth in the course of valida-

tion and execution, depending on the addressed situation (to-be or as-it-is).

5. Organizational validation and execution is best achieved involving role carriers when probing models through executing them. It fosters a lively engagement when aiming to find organization-wide consensus along interactive experiences of process models.

Cross-checking the presented validation and implementation techniques, each of the presented technique focus on role-specific behavior and their interaction patterns. Scaffolding extends the variability of how to align and validate knowledge for an organizational consolidation and, finally, implementations. Opening up for innovative or novel process design is considered particularly useful in case of highly complex situations or conflicting perspectives on workflows. Scaffolds also support middle-out development of organizational structures, as they enable multilayered perspectives and bootstrap alignment processes. The latter is supported only implicitly by going back and forth in subject-oriented modeling coupled with interactive prototyping of executable process models.

References

- Abdu, Rotem, Baruch Schwarz, and Manolis Mavrikis. 2015. Whole-Class Scaffolding for Learning to Solve Mathematics Problems Together in a Computer-Supported Environment. *Zdm* 47 (7): 1163–1178 (Springer).
- Arias, E.G., and G. Fischer. 2000. Boundary Objects: Their Role in Articulating the Task at Hand and Making Information Relevant to It. In *Proceedings of the International ICSC Symposium on Interactive and Collaborative Computing (ICC'2000)*.
- Arias, Ernesto, H. Eden, Gerhard Fischer, A. Gorman, and E. Scharff. 2000. Transcending the Individual Human Mind—Creating Shared Understanding Through Collaborative Design. *ACM Transactions on Computer-Human Interaction (TOCHI)* 7 (1): 84–113 (New York: ACM Press).
- Auinger, Andreas, and C. Stary. 2005. *Didaktikgeleiteter Wissenstransfer. Interaktive Informationsräume Für Lern-Gemeinschaften Im Web*. Deutscher Universitätsverlag.

- Berztiss, Alfs. 1996. Business Process Prototyping. In *Software Methods for Business Reengineering*, 122–125. Springer.
- Beyer, H., and K. Holtzblatt. 1997. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann.
- Briggs, Robert O., G.L. Kolfschoten, Gert Jan De Vreede, Stephan Lukosch, and Conan C. Albrecht. 2013. Facilitator-in-a-Box: Process Support Applications to Help Practitioners Realize the Potential of Collaboration Technology. *Journal of Management Information Systems* 29 (4): 159–194 (Taylor & Francis).
- Bruno, G., F. Dengler, B. Jennings, R. Khalaf, S. Nurcan, M. Prilla, Marcello Sarini, R. Schmidt, and R. Silva. 2011. Key Challenges for Enabling Agile BPM with Social Software. *Journal of Software Maintenance and Evolution: Research and Practice* 23 (4): 297–326 (Wiley Online Library).
- Bulu, Saniye Tugba, and Susan Pedersen. 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): 507–529 (Springer).
- Caporale, T. 2016. A Tool for Natural Language Oriented Business Process Modeling. In *Proceedings of ZEUS 2016*, 49. ceur-ws.org.
- Chin, Christine, and Jonathan Osborne. 2010. Students' Questions and Discursive Interaction: Their Impact on Argumentation During Collaborative Group Discussions in Science. *Journal of Research in Science Teaching* 47 (7): 883–908 (Wiley Online Library).
- Convertino, Gregorio, Helena M. Mentis, Mary Beth Rosson, John M. Carroll, Aleksandra Slavkovic, and Craig H. Ganoe. 2008. Articulating Common Ground in Cooperative Work: Content and Process. In *CHI '08: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems*, 1637–1646. New York: ACM. <https://doi.org/10.1145/1357054.1357310>.
- Dalmaris, Peter, Eric Tsui, Bill Hall, and Bob Smith. 2007. A Framework for the Improvement of Knowledge-Intensive Business Processes. *Business Process Management Journal* 13 (2): 279–305 (Emerald).
- Davies, Islay, Peter Green, Michael Rosemann, Marta Indulska, and Stan Gallo. 2006. How Do Practitioners Use Conceptual Modeling in Practice? *Data & Knowledge Engineering* 58 (3): 358–380.
- Dennen, Vanessa Paz. 2004. Cognitive Apprenticeship in Educational Practice: Research on Scaffolding, Modeling, Mentoring, and Coaching as Instructional Strategies. In *Handbook of Research on Educational Communications and Technology*, 2nd ed., 813–828. Mahwah, NJ: Lawrence Erlbaum Associates.

- Dittmar, A., P. Forbrig, S. Heftberger, and C. Stry. 2004. Support for Task Modeling—A “Constructive” Exploration. In *International Workshop on Design, Specification, and Verification of Interactive Systems*, 59–76. Berlin, Heidelberg: Springer.
- Ellson, J., E. Gansner, L. Koutsofios, S.C. North, and G. Woodhull. 2001. Graphviz—Open Source Graph Drawing Tools. In *International Symposium on Graph Drawing*, 483–484. Berlin, Heidelberg: Springer.
- Firestone, J.M., and M.W. McElroy. 2003. *Key Issues in the New Knowledge Management*. Butterworth-Heinemann.
- Fleischmann, Albert, Werner Schmidt, C. Stry, Stefan Obermeier, and Egon Börger. 2012. *Subject-Oriented Business Process Management*. Springer.
- Floch, Jacqueline, Svein Hallsteinsen, Erlend Stav, Frank Eliassen, Ketil Lund, and Eli Gjorven. 2006. Using Architecture Models for Runtime Adaptability. *IEEE Software* 23 (2): 62–70 (IEEE).
- Floyd, Christiane. 1984. A Systematic Look at Prototyping. In *Approaches to Prototyping*, 1–18. Springer.
- Forster, Simon, Jakob Pinggera, and Barbara Weber. 2013. Toward an Understanding of the Collaborative Process of Process Modeling. *CAiSE Forum*, 98–105.
- Franco, L. Alberto. 2013. Rethinking Soft or Interventions: Models as Boundary Objects. *European Journal of Operational Research* 231 (3): 720–733 (Elsevier).
- Franco, L. Alberto, and Étienne A.J.A. Rouwette. 2011. Decision Development in Facilitated Modelling Workshops. *European Journal of Operational Research* 212 (1): 164–178. <https://doi.org/10.1016/j.ejor.2011.01.039>.
- Frederiks, P.J.M., and Th.P. van der Weide. 2006. Information Modeling: The Process and the Required Competencies of Its Participants. *Data & Knowledge Engineering* 58 (1): 4–20. <https://doi.org/10.1016/j.datak.2005.05.007>.
- Front, A., D. Rieu, M. Santorum, and F. Movahedian. 2017. A Participative End-User Method for Multi-Perspective Business Process Elicitation and Improvement. *Software & Systems Modeling* 16 (3): 691–714.
- Ge, Xun, and Susan M. Land. 2004. A Conceptual Framework for Scaffolding III-Structured Problem-Solving Processes Using Question Prompts and Peer Interactions. *Educational Technology Research and Development* 52 (2): 5–22 (Springer).
- Hartmann, B., L. Yu, A. Allison, Y. Yang, and S.R. Klemmer. 2008. Design as Exploration: Creating Interface Alternatives Through Parallel Authoring and Runtime Tuning. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, 91–100. ACM.

- Herrmann, T., and K.-U. Loser. 2013. Facilitating and Prompting of Collaborative Reflection of Process Models. In *Proceedings of MoRoCo@ECSCW 2013*, 17–24. ceur-ws.org.
- Herrmann, Thomas, G. Kunau, K.U. Loser, and N. Menold. 2004. Socio-Technical Walkthrough: Designing Technology Along Work Processes. In *Artful Integration: Interweaving Media, Materials and Practices. Proceedings of the Eighth Conference on Participatory Design*, 132–141. New York: ACM Press.
- Herrmann, Thomas, K.U. Loser, and I. Jahnke. 2007. Sociotechnical Walkthrough: A Means for Knowledge Integration. *The Learning Organization* 14 (5): 450–464.
- Herrmann, Thomas, M. Hoffmann, G. Kunau, and K.U. Loser. 2002. Modelling Cooperative Work: Chances and Risks of Structuring. In *Cooperative Systems Design, a Challenge of the Mobility Age. Proceedings of COOP 2002*, 53–70. IOS Press.
- Hjalmarsson, Anders, Jan C. Recker, Michael Rosemann, and Mikael Lind. 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): 421–447.
- Holbrook, H. 1990. A Scenario-Based Methodology for Conducting Requirements Elicitation. *ACM SIGSOFT Software Engineering Notes* 15 (1): 95–104. <https://doi.org/10.1145/382294.382725>.
- Jumaat, N.F., and Z. Tasir. 2014, April. Instructional Scaffolding in Online Learning Environment: A Meta-analysis. In *2014 International Conference on Teaching and Learning in Computing and Engineering*, 74–77. IEEE.
- Kensing, Finn, and Jeanette Blomberg. 1998. Participatory Design: Issues and Concerns. *Computer Supported Cooperative Work (CSCW)* 7 (3–4): 167–185 (Springer).
- King, Alison, and Barak Rosenshine. 1993. Effects of Guided Cooperative Questioning on Children's Knowledge Construction. *The Journal of Experimental Education* 61 (2): 127–148 (Taylor & Francis).
- Kobbe, Lars, Armin Weinberger, Pierre Dillenbourg, Andreas Harrer, Raija Hämäläinen, Päivi Häkkinen, and Frank Fischer. 2007. Specifying Computer-Supported Collaboration Scripts. *International Journal of Computer-Supported Collaborative Learning* 2 (2–3): 211–224 (Springer).
- Krenn, Florian, and Christian Stary. 2016. Exploring the Potential of Dynamic Perspective Taking on Business Processes. *Complex Systems Informatics and Modeling Quarterly*, no. 8: 15–27. <https://csimq-journals.rtu.lv/issue/view/87>.
- Land, Susan M., and Carla Zembal-Saul. 2003. Scaffolding Reflection and Articulation of Scientific Explanations in a Data-Rich, Project-Based

- Learning Environment: An Investigation of Progress Portfolio. *Educational Technology Research and Development* 51 (4): 65–84 (Springer).
- Mayer, R.E. 1989. Models for Understanding. *Review of Educational Research* 59 (1): 43–64.
- Minor, M., A. Tartakovski, and D. Schmalen. 2008. Agile Workflow Technology and Case-Based Change Reuse for Long-Term Processes. *International Journal of Intelligent Information Technologies (IJIIT)* 4 (1): 80–98 (IGI Global).
- Monsalve, C., A. April, and A. Abran. 2010. Representing Unique Stakeholder Perspectives in BPM Notations. In *2010 Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA)*, 42–49.
- Mori, G., F. Paternò, and C. Santoro. 2002. CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. *IEEE Transactions on Software Engineering* 28 (9): 797–813.
- Morris, W.T. 1967. On the Art of Modeling. *Management Science* 13 (12): B-707–B-717. <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.13.12.B707>.
- Neubauer, Matthias, Stefan Oppl, C. Stary, and Georg Weichhart. 2013. Facilitating Knowledge Transfer in IANES—A Transactive Memory Approach. In *Innovation Through Knowledge Transfer 2012*, Smart Innovation, Systems and Technologies, vol. 18, ed. R. Howlett, B. Gabrys, K. Musial-Gabrys, and J. Roach, 39–50. Berlin and Heidelberg: Springer. https://doi.org/10.1007/978-3-642-34219-6_5.
- Neubauer, Matthias, C. Stary, and Stefan Oppl. 2011. Polymorph Navigation Utilizing Domain-Specific Metadata: Experienced Benefits for E-Learners. In *Proceedings of the 29th European Conference on Cognitive Ergonomics (ECCE 2011)*, 45–52. ACM Press.
- Nielsen, Jakob. 1993. Iterative User-Interface Design. *Computer* 26 (11): 32–41 (IEEE).
- Nolte, Alexander, and M. Prilla. 2012. Normal Users Cooperating on Process Models: Is It Possible at All? In *Collaboration and Technology*, 57–72. Berlin: Springer.
- Oppl, Stefan. 2015. Articulation of Subject-Oriented Business Process Models. In *Proceedings of S-BPM ONE 2015*, 1–11. New York: ACM Press. <https://doi.org/10.1145/2723839.2723841>.
- . 2016. Articulation of Work Process Models for Organizational Alignment and Informed Information System Design. *Information & Management* 53 (5): 591–608. <https://doi.org/10.1016/j.im.2016.01.004>.
- Oppl, Stefan, and Nancy Alexopoulou. 2016. Linking Natural Modeling to Techno-Centric Modeling for the Active Involvement of Process Participants in Business Process Design. *International Journal of Information System*

- Modeling and Design* 7 (2): 1–30. <https://doi.org/10.4018/IJISMD.2016040101>.
- Ozmantar, M.F., and T. Roper. 2004. Mathematical Abstraction Through Scaffolding. In *Proceedings of the 28th Conference of the International Group for the Psychology of Mathematics Education*, vol. 3, 481–488.
- Pinelle, D., and C. Gutwin. 2002, April. Groupware Walkthrough: Adding Context to Groupware Usability Evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 455–462. ACM.
- Pirnay-Dummer, Pablo N., and A. Lachner. 2008. Towards Model Based Knowledge Management. A New Approach to the Assessment and Development of Organizational Knowledge. In *Annual Proceedings of the AECT 2008*, ed. M. Simonson, 178–118.
- Polson, Peter G., Clayton Lewis, John Rieman, and Cathleen Wharton. 1992. Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Studies* 36 (5): 741–773 (Elsevier).
- Powell, S.G., and T. R. Willemain. 2007. How Novices Formulate Models. Part I: Qualitative Insights and Implications for Teaching. *Journal of the Operational Research Society*, 58: 983–995 (JSTOR).
- Prilla, Michael. 2015. Supporting Collaborative Reflection at Work: A Socio-Technical Analysis. *AIS Transactions on Human-Computer Interaction* 7 (1): 1–17.
- Prilla, M., and Alexander Nolte. 2012. Integrating Ordinary Users into Process Management: Towards Implementing Bottom-Up, People-Centric BPM. In *Enterprise, Business-Process and Information Systems Modeling*, 182–194. Springer.
- Prusak, Naomi, Rina Hershkowitz, and Baruch B. Schwarz. 2012. From Visual Reasoning to Logical Necessity Through Argumentative Design. *Educational Studies in Mathematics* 79 (1): 19–40 (Springer).
- Recker, J.C., Norizan Safrudin, and Michael Rosemann. 2012. How Novices Design Business Processes. *Information Systems* 37 (6): 557–573 (Elsevier Science Ltd).
- Reichert, Manfred, and Peter Dadam. 2009. Enabling Adaptive Process-Aware Information Systems with ADEPT2. In *Handbook of Research on Business Process Modeling*, 173–203. IGI Global.
- Roberts, Andrew. 2009. Encouraging Reflective Practice in Periods of Professional Workplace Experience: The Development of a Conceptual Model. *Reflective Practice* 10 (5): 633–644 (Taylor & Francis).
- Sandkuhl, Kurt, Janis Stirna, Anne Persson, and Matthias Wißotzki. 2014. Elicitation Approaches in Enterprise Modeling. In *Enterprise Modeling*, The

- Enterprise Engineering Series, 39–51. Berlin and Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-43725-4_4.
- Santoro, Flávia Maria, Marcos R.S. Borges, and José A. Pino. 2010. Acquiring Knowledge on Business Processes from Stakeholders' Stories. *Advanced Engineering Informatics* 24 (2): 138–148. <https://doi.org/10.1016/j.aei.2009.07.002>.
- Schiffner, S., T. Rothschild, and N. Meyer. 2014, September. Towards a Subject-Oriented Evolutionary Business Information System. In *2014 IEEE 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations*, 381–388. IEEE.
- Smeds, Riitta, and Jukka Alvesalo. 2003. Global Business Process Development in a Virtual Community of Practice. *Production Planning & Control* 14 (4): 361–371 (Taylor & Francis).
- Sousa, Kênia, Hildeberto Mendonça, Amandine Lievyens, and Jean Vanderdonckt. 2011. Getting Users Involved in Aligning Their Needs with Business Processes Models and Systems. *Business Process Management Journal* 17 (5): 748–786 (Emerald Group Publishing Limited).
- Stender, Peter, and Gabriele Kaiser. 2015. Scaffolding in Complex Modelling Situations. *Zdm* 47 (7): 1255–1267 (Springer).
- Su, J.-M. 2015. A Self-regulated Learning Tutor to Adaptively Scaffold the Personalized Learning: A Study on Learning Outcome for Grade 8 Mathematics. In *2015 8th International Conference—Presented at the Ubi-Media Computing (UMEDIA)*, 376–380.
- Sukaviriya, N., V. Sinha, T. Ramachandra, S. Mani, and M. Stolze. 2007, September. User-Centered Design and Business Process Modeling: Cross Road in Rapid Prototyping Tools. In *IFIP Conference on Human-Computer Interaction*, 165–178. Berlin, Heidelberg: Springer.
- Tavella, Elena, and Thanos Papadopoulos. 2014. Expert and Novice Facilitated Modelling: A Case of a Viable System Model Workshop in a Local Food Network. *Journal of the Operational Research Society* 66 (2): 247–264 (Palgrave Macmillan).
- Van de Pol, Janneke, Monique Volman, and Jos Beishuizen. 2010. Scaffolding in Teacher—Student Interaction: A Decade of Research. *Educational Psychology Review* 22 (3): 271–296 (Springer).
- Vogel, F., C. Wecker, I. Kollar, and F. Fischer. 2017. Socio-Cognitive Scaffolding with Computer-Supported Collaboration Scripts: A Meta-analysis. *Educational Psychology Review* 29 (3): 477–511.
- Vygotsky, L.S. 1978. *Mind in Society*. Harvard University Press.
- Wachholder, Dominik, and Stefan Oppl. 2012. Stakeholder-Driven Collaborative Modeling of Subject-Oriented Business Processes. In *S-BPM*

- ONE—*Scientific Research*, ed. C. Stary, 145–162. Berlin and Heidelberg: Springer. https://doi.org/10.1007/978-3-642-29133-3_10.
- . 2014. Interactive Coupling of Process Models: A Distributed Tabletop Approach to Collaborative Modeling. In *ECCE '14: Proceedings of the 2014 European Conference on Cognitive Ergonomics*, September, 1–8. New York: ACM Request Permissions. <https://doi.org/10.1145/2637248.2637262>.
- Walls, Joseph G., George R. Widmeyer, and Omar A. El Sawy. 1992. Building an Information System Design Theory for Vigilant EIS. *Information Systems Research* 3 (1): 36–59 (Informs).
- Weske, Mathias. 2010. *Business Process Management: Concepts, Languages, Architectures*. Springer.
- Willemain, Thomas R. 1995. Model Formulation: What Experts Think About and When. *Operations Research* 43 (6): 916–932 (Informs).
- Wood, David, Jerome S. Bruner, and Gail Ross. 1976. The Role of Tutoring in Problem Solving. *Journal of Child Psychology and Psychiatry* 17 (2): 89–100 (Wiley Online Library).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

