Wiki-Based Maturing of Process Descriptions

Frank Dengler and Denny Vrandečić

Karlsruhe Institute of Technology (KIT),
Institute AIFB, 76128 Karlsruhe, Germany
{frank.dengler,denny.vrandecic}@kit.edu

Abstract. Traditional process elicitation methods are expensive and time consuming. Recently, a trend toward collaborative, user-centric, online business process modeling can be observed. Current social software approaches, satisfying such a collaborative modeling, mostly focus on the graphical development of processes and do not consider existing textual process description like HowTos or guidelines. We address this issue by combining graphical process modeling techniques with a wiki-based light-weight knowledge capturing approach and a background semantic knowledge base. Our approach enables the collaborative maturing of process descriptions with a graphical representation, formal semantic annotations, and natural language. Existing textual process descriptions can be translated into graphical descriptions and formal semantic annotations. Thus, the textual and graphical process descriptions are made explicit and can be further processed. As a result, we provide a holistic approach for collaborative process development that is designed to foster knowledge reuse and maturing within the system.

1 Introduction

Enterprises are trying to describe their business processes in order to better understand, share, and optimize them. But still most of the process knowledge remains either in people's heads, or as textual and graphical descriptions in the Intranet as how-tos, guidelines, or methodology descriptions. The cost of an upfront, complete formalization of all business processes is prohibitive, and the benefits often seem elusive, especially under the stress of the daily work.

Traditionally, interviews of the domain experts performed by process modelers have been used to develop the process descriptions. An alternative to interviews is the group storytelling approach. It transforms stories told by individual process performers into process descriptions [23].

In this paper we present a process and a wiki-based platform that allows capturing process descriptions through several channels, with different speeds and goals. It supports the capturing of stories and natural language process descriptions, rendering and editing of graphical representations, and formal models, which can be exported with a well-defined semantics and used for the further processing and validation. The platform covers a wide array of externalization approaches, and provides a unifying system that can interconnect gradually the explicated knowledge, enabling the enterprise to follow a continuous knowledge

S. Rinderle-Ma, F. Toumani, and K. Wolf (Eds.): BPM 2011, LNCS 6896, pp. 313–328, 2011.

maturing effort [25,24] instead of requiring a steep and complete formalization step.

We have implemented a fully functional prototype[1] where users can start to model their processes by formulating a first idea with natural language or with an incomplete model. These models can be further refined and consolidated by others. Existing processes within the wiki platform either formatted as a textual how-to or expressed via predefined semantic properties can be reused by transforming them into a newly developed graphical representation. We have evaluated our approach by conducting functionality tests and a usability study to check if users can graphically render and edit processes in an intuitive manner.

The paper is structured as follows. Section 2 describes the requirements for collaborative process development derived from previous literature and an analysis of existing processes in industry. As a result of our requirement analysis, we present our approach to support collaborative process development (Section 3) and its implementation in Section 4. After describing our holistic approach for collaborative proposal development, we elaborate the transformation of existing how-tos into graphical process representations in Section 5. We evaluate our approach by modeling and transforming existing processes in Section 6. Section 7 relates the elaborations of this paper to other research. Eventually, Section 8 concludes the paper by mapping the requirements on the features of the implementation, and gives an outlook on future research.

2 Requirement Analysis

In order to collect the requirements for enabling both novice users of our system as well as process modeling experts to gather process descriptions and cooperate together, we analyzed previous literature and further requirements derived from our work with case study partners. We present and summarize these requirements in this section.

2.1 Requirements from Literature

In order to reuse, share, and collaboratively improve processes, they have to be externalized. Often, process mining techniques and tools [1] are used to capture and analyze activities performed by knowledge workers. But many tasks of knowledge workers are outside the grasp of mining algorithms: more than 14% of the work time of knowledge workers is spent in phone calls or meetings (as shown by a study of task switching and interruptions of knowledge workers [5]), which today is not sufficiently monitored to be a viable source for process mining. Knowledge required for process activities is often exchanged informally between knowledge workers, e.g. during a discussion at the water cooler or over lunch, and thus also out of the grasp of automatic approaches.

[1] The prototype can be accessed via `http://bpmexample.wikiing.de` (User name: `ProcessTester` – Password: `active!`).

Complementary to mining the activities of the knowledge worker, process modeling tools can be used to explicitly capture the processes. Knowledge workers can model their own processes, or simply tell their own stories. The aggregated process descriptions and stories can be used to arrive at a high-quality description of the actually performed processes. Thus, collaboration support is required. People with different levels of expertise are describing processes. Therefore it is equally important to provide means for the individual knowledge worker to externalize their process knowledge, as well as for process modeling experts to efficiently work on the abstraction level they are used. The integration of stakeholders into the creation, adaptation, and revision of process models has been shown beneficial especially for model accuracy and verification [15]. This requires the tool to enable to bridge the gap between the modeling expertise levels of the knowledge workers and the modeling experts.

Recker et al. [21] have investigated how novices model business processes in the absence of tool support. They have found that design representation forms chosen to conceptualize business processes range from predominantly textual over hybrid to predominantly graphical types. The hybrid process descriptions, combining graphical and textual types, achieve a higher quality.

A survey analyzing the used modeling constructs of the Business Process Modeling Notation[2] (BPMN), shows that in most BPMN diagrams less than 20% of the BPMN vocabulary constructs are regularly used and the most occurring subset is the combination of *tasks* and *sequence flows* [19]. A simpler subset of the language can be chosen based on this results.

2.2 Requirements from Use Cases

For our analysis of processes used in industry we looked at twenty different processes from a large consultancy company. The processes are methodologies and reusable assets describing procedures to guide consultants, i.e. knowledge workers, in their daily work. The processes are defined by experts, but junior consultants with only little experience in process modeling have to work with them. The older process descriptions are described in MS Word documents, accessible from the company Intranet. The newer ones are stored as Flash and HTML files on the Intranet. A process can have subprocesses which are stored in separate files and interrelated to each other via links. Each process document contains a short description, inputs and outputs, a flow diagram, and extensive textual descriptions of each process step. The process flow is expressed by a process picture created with a graphics software (i.e. the semantics of the picture is not formally captured and accessible to the machine for further processing). The textual descriptions are composed of detailed action instructions, links to other resources, and the roles responsible for each step. All (sub)processes contain less than 10 steps. Only a few modeling constructs are used within the flow models, namely *activities*, *sequence flows*, *conditions*, and *pools*. Instead of using BPMN to capture the more intricate details of the processes, the expressivity of the

[2] `http://www.bpmn.org/`

flow model is complemented by textual descriptions, e.g. exception handling is described in detail within the action instructions. Due to that setup, the process search is limited to a simple textual search over the descriptions as the process knowledge is not machine-processable and can not be exploited for the search. Browsing is also confined to the few links provided explicitly by the authors of the process descriptions. To increase the search and navigation functionality, support for structured process documentation is required.

2.3 Summary of Requirements

Based on the previous two sections, we derive the following requirements for collaborative process development:

Natural Language Support for Novice Users. Novices in process modeling need to be able to use natural language instead of formal modeling constructs, so that they can create and extend the processes without the assistance of an expert. If the user does not know the graphical representation of a process element, natural language can be used to describe it.

Intuitive Graphical Rendering and Editing of Processes. A rich user interface can provide the user with means for interacting with processes in a highly intuitive manner. But with too many elements, the usability of the tool may suffer. This leads to a trade-off between the expressivity offered to develop the formal process models and the simple accessibility of the tool.

Collaboration Support. Users must be able to discuss process models asynchronously. Changes of the process model have to be tracked and users should be enabled to access the version history and to revert to previous versions. In addition, design rationales should be documented.

Structured Process Documentation Support. The process models must be stored in a machine-processable documentation format, including additional properties linking to external resources. Users must be able to interlink between process descriptions and external resources to enable more sophisticated retrieval, browsing, and navigation.

3 Approach

To address these requirements for collaborative process development, we have combined wiki-based light-weight knowledge capturing with graphical process modeling functionality and a semantic knowledge base. Hence, users can develop process descriptions by using graphical representations, natural language, and formal semantic annotations.

Our approach enables domain experts without modeling expertise to actively participate in the process modeling task. It provides users with means to intuitively model processes graphically with basic (but widely used) process elements, namely *task*, *sequence flow*, *parallel gateway*, and *data-based exclusive gateway*. This allows to integrate stakeholders into the creation, adaption, and revision of the process descriptions. A single person typically does not have a complete overview of a process. The required organizational and semantic integration are also addressed with our approach; all users can contribute to process modeling and the definition of terms is done in a multitude of small steps, giving each contributor the ability to adapt easily to a common usage of terms [4]. Standard wiki features, like the discussion functionality, also support the definition of terms in a collaborative manner. Thus, users can discuss and evolve the meaning of terms. By using the watch list functionality, users can be notified about changes in the process descriptions or new contributions to the discussion.

The process models are stored in a machine-processable format. These are created on the one hand by automatically translating the graphical process representation into formal semantic annotations, but also by using the annotations from the semantic wiki. During the translation of the graphical representation a wiki page is created for each process element with corresponding semantic properties. A predefined property is introduced for the sequence flow. In addition, the user can enter annotated links to external resources that are required for a given step or task (e.g assigned roles and actors). The formal process descriptions are the combined results from the graphics and the annotated text. Using the formal descriptions, a semantic search system is offered that can improve the navigation and retrieval of process activities. In addition, semantic queries can be used to detect errors or constraint violations in process model descriptions, e.g. a search for the executors of all process activities using a specific confidential document.

An application of our approach can be a collaborative proposal development scenario. Since many people with different expertise and roles are involved, everybody can contribute in acquiring the proposal development processes. Team members can start modeling from scratch or by translating existing textual process descriptions. Depending on the expertise and the available time, proposal team members can thereby alter the process to take advantage of it (e.g. contributing only to parts of the proposal document for which they have the necessary knowledge). Additionally, previous proposal documents can be linked with our approach to specific process activities using semantic properties. Thus relevant proposals can be filtered out by querying for related, previously created proposals. Also less experienced proposal team members can profit from the process wiki, because they can look up and follow the developed processes. The formalized processes can also be used as a basis for the input to a process execution engine, e.g. by accessing the RDF export interface of our approach via the process execution tool.

4 Implementation

We extended our implementation presented in [8] and integrated a graphical process editor with Semantic MediaWiki (SMW). SMW extends the MediaWiki [2] software that runs, e.g. the popular Wikipedia site. SMW combines Semantic Web technology with the collaborative aspects of wikis [17] to enable large-scale and inter-departmental collaboration on knowledge structures. Knowledge can be expressed by using natural language in combination with formal annotations allowing machines to process this knowledge. An extension to the MediaWiki syntax is provided by SMW to define class and property hierarchies, and semantic properties related to wiki pages. For instance a category *Process* can be added to a wiki page by adding [[Category:Process]] on the wiki page. For a property *has Successor*, expressing a successor relation between two wiki pages, the following syntax is used: [[has Successor::<Name of wiki page>]].

To access the formalized knowledge within wiki pages, SMW offers an inline query language (ASK syntax). The syntax for a query asking for all instances belonging to the category *Process* and their property *Short Description* is {{#ask: [[Category:Process]] |?Short Description}}. Without stating a specific output format, the query result will be displayed by default as a table on the corresponding wiki page. To make the formalized knowledge also available for other applications, SMW provides several export functionalities, e.g. in RDF [3].

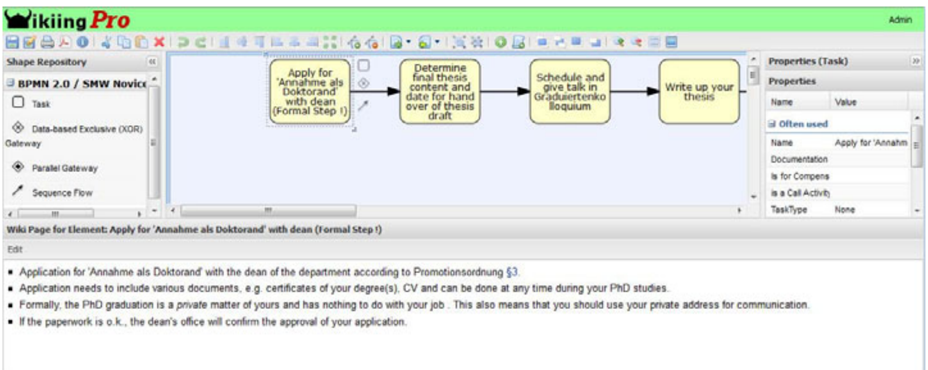

Fig. 1. SMW Process Editor Screen Shot

For our implementation we selected the Oryx Process Editor [6], an open source process editor, as the graphical process editor component. It currently supports various modeling languages such as BPMN, Event-driven Process Chain (EPC), Petri Nets, Workflow Nets, as well as Unified Modeling Language (UML), and can easily be extended to handle further process modeling languages.

SMW was extended to be compatible with the Oryx graphical editor, so that data can be exchanged between both. In addition, the Oryx graphical editor was extended to display and edit wiki pages seamlessly from within its interface; as a consequence, users can directly access the corresponding wiki page within the process editor. The entered wiki text is rendered by using the parse method provided by SMW. Thus the whole SMW syntax can be used including categories and properties. SMW ASK queries are executed and the results are displayed as well. Both the originally entered text as well as the parsed wiki text are temporarily stored within the data model of the process editor as additional hidden properties.

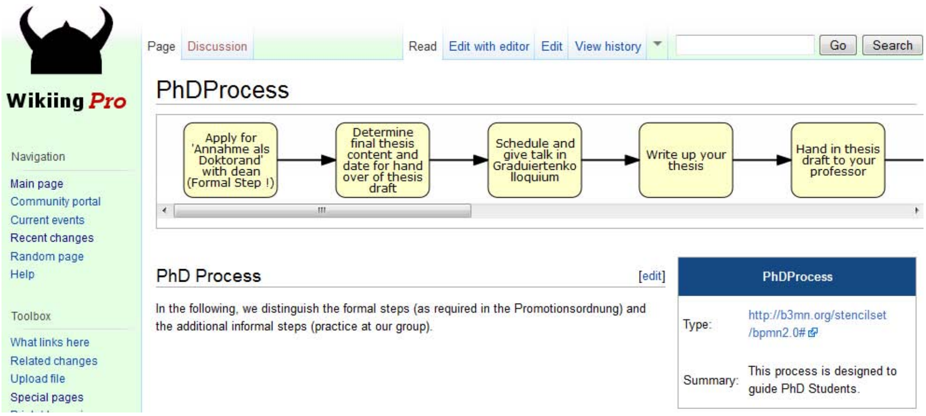

Fig. 2. Example Process in SMW (process summary page with fact box)

As can be seen in Figure 1, the process editor interface consists of different regions. The corresponding wiki page is displayed in the bottom of the editor. For our approach we only use a small subset of BPMN constructs. The available process elements are presented in the left region of the editor, namely *task*, *sequence flow*, *parallel gateway*, and *data-based exclusive gateway*. Users can easily add process elements to the process by dragging a process element from the left region and dropping it on the process diagram in the middle. Once the process is saved (by clicking on the save button in the process editor), the process data and wiki pages belonging to the process are created or updated in SMW. The process elements are saved as subpages to the process summary page within the wiki (see Figure 3 for an example). The process element wiki pages contain the textual descriptions and a fact box with all the stored properties. On the process summary page, the process diagram in SVG and a fact box are displayed (see Figure 2 for an example).

We support most of the *Basic Control-Flow Patterns* introduced in [22]. Every single process step (activity) is represented as a wiki page belonging to the category *Process Element* and linked via the properties *has Type* to the corresponding type (*task*) and *Belongs to Process* to the corresponding process, represented as wiki pages themselves (process summary pages). An activity is the basic element of our process. Depending on the granularity level of the process this can vary from atomic activities, such as *open a web page*, to activities describing a whole subprocess. To express the control flow of the process, we use edges in the diagram and special predefined process elements (gateways). If an element has a successor, we draw an edge from the activity to the successor activity in the diagram and store this with the additional property *has Successor* on the corresponding wiki page in SMW. For more successors executed in parallel (parallel-split pattern), a *parallel gateway* is used in between the activities. An activity can have several successors, but only one has to be selected and executed (multi-choice pattern). Therefore we use the *data-based exclusive gateway*

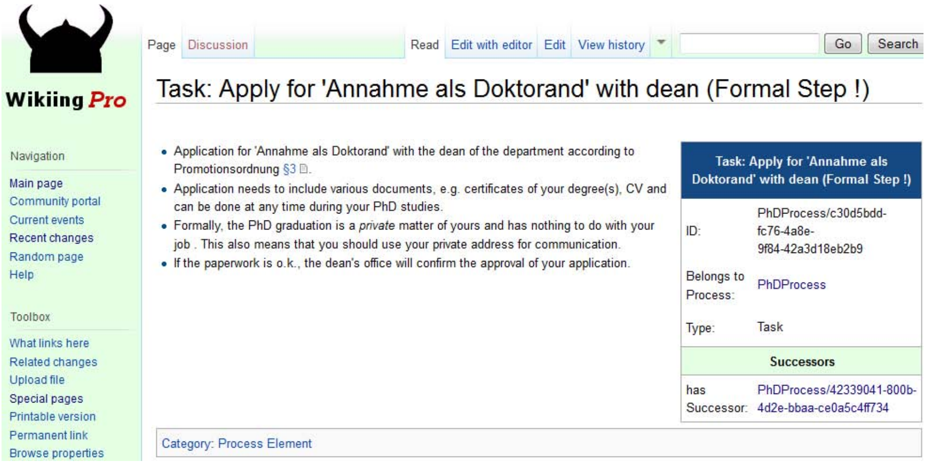

Fig. 3. Example Task in SMW (element wiki page with fact box)

without conditions. The *data-based exclusive gateway* with conditions is used to split based on a condition (exclusive-choice pattern). A condition is stored as a many-valued property[3]. The distinction between the synchronization pattern and the simple-merge pattern is realized by using the *parallel gateway* and the *data-based exclusive gateway* the other way round to merge different branches of a process.

All properties of the process elements are also available in SMW. They are stored as SMW properties with their corresponding value. Thus all the process properties can be accessed within SMW, queried, and exported. The basic vocabulary used in the wiki is illustrated in Figure 4. For example, these properties can be displayed in a fact box on the corresponding wiki page as shown in Figure 2. Links to the corresponding wiki pages are automatically added to the SVG graphic, which enable the user to navigate through the process in the wiki.

A new tab *edit with editor* has been added to the process wiki for editing an existing process. The tab automatically appears on pages belonging to the categories *Process* and *Process Element*. The tab contains the graphical process editor with the process model. Even though technically the process editor runs on a separate Tomcat server, the SMW authentication is also used to control the access on the process model within the process editor, providing a seamless experience between the two different components.

5 Transformation of Existing Textual Processes

In traditional (semantic) wikis, processes are stored either as a textual description on a single wiki page or they can also be expressed by connecting various

[3] Many-valued properties in SMW are implemented as records, see `http://semantic-mediawiki.org/wiki/Type:Record`

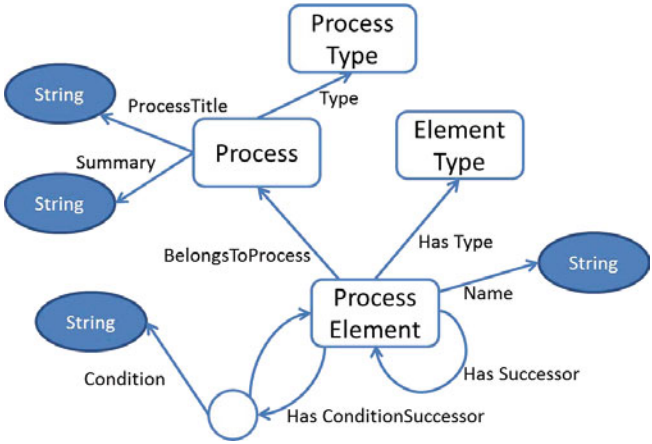

Fig. 4. Basic Wiki Process Vocabulary

wiki pages. The single page process descriptions are usually simple HowTos, structured as a numbered list. Examples for such how-tos can be found at WikiHow[4], a wiki platform where users can collaboratively create, edit, and share how-tos. The interlinking of multiple wiki pages can define a single process (by describing single tasks or subprocesses on their own pages, and then interlinking these). A way to specify such a process, by using predefined semantic properties to express flow relations, is described in [8]. With our wiki-based approach, we support both ways of transforming wiki pages into graphically editable process descriptions – single and multi wiki page transformations.

As shown in Figure 5, a *Create Process* tab is displayed on each wiki page that does not belong already to a graphical process description. By clicking on this tab, the graphical editor interface is displayed and the described process is transformed as described in the following subsections.

5.1 Single Wiki Page Transforming

Simple how-tos, like in WikiHow, are usually formatted as numbered lists. Text within MediaWiki can be formatted as enumerations, or ordered lists. MediaWiki syntax allows building ordered lists by using the hash character (`#`) at the beginning of subsequent lines[5]. For the transformation of such a single page how-to, our extension parses the text for the hash character at the beginning of a line, similar to our previous approach in [26]. If a text line matches this pattern, we create a task element and display the text either as the name of the process step or on the corresponding wiki page, if it is longer than 100 characters. Bullet lists and definition items, included in the numbered list, are directly displayed as part of the corresponding wiki page.

[4] `http://www.wikihow.com`

[5] `http://www.mediawiki.org/wiki/Help:Formatting`

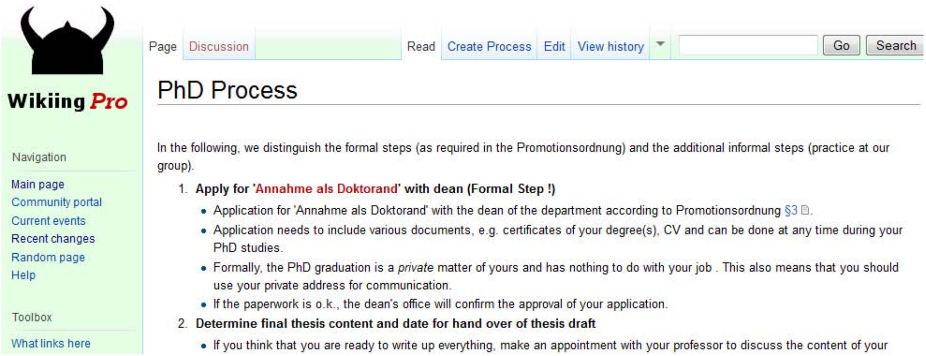

Fig. 5. Example How-to in SMW

A how-to can look like the numbered list shown in Figure 5. The graphical transformation of this textual process description is illustrated in Figure 1. Upon saving the page, the enumeration will be replaced with the process summary page as shown in Figure 2.

5.2 Multi Wiki Page Transforming

For transforming multiple wiki pages describing a process, where the pages are connected with flow relations, we used the predefined properties specified in [8] to express the process flow. A *task* element is created in the process diagram for each single wiki page, retrieved with the appropriate SMW query[6]. For each property `has Successor` an edge (*sequence flow* element) is created between the subject and object of the property. If a *task* has multiple successor properties, a *parallel gateway* element is included, from where the edges go to the multiple successor tasks. The *data-based exclusive gateway* is used for each `has OrSuccessor` property and their condition properties. The content of the wiki page defining the process is also included in the corresponding process steps, thus providing an overview.

When the user saves the transformed graphical representation of the process description, the process in the new structure is stored in the wiki. A redirect link is inserted on each previous element of the process description to the corresponding new process step wiki page. With these redirects in place we can assure minimal negative impact due to the transformation and allow all existing links to continue working.

6 Evaluation

We evaluated our holistic approach for collaborative maturing of process descriptions in three steps built upon each other.

[6] `[[Belongs to Process::<Page name>]].`

1. We applied SMW with the collaborative process development functionality presented in [8] to industry use cases within the ACTIVE project[7]. This allowed for the creation of process descriptions using merely the textual, wiki-based syntax. No specialized or graphical user interface was available for editing the processes. After the trialists started to use the process visualization functionality (as query results, i.e. an uneditable graph displaying the process), they suggested that the process-editing functionality should be easier to use [9], e.g. similar to Microsoft Visio.
2. Therefore, our new approach presented in this paper supports the graphical editing of processes. The usability of the wiki-based graphical process editor was evaluated with ten students, modeling textual process descriptions. We used four textual process descriptions specifying internal HowTos of the university institute AIFB and eight service process descriptions (GR01, GR02, GR03, GR04, GR05, IT01, IT02, and IT04) from the COCKPIT Project [16]. The processes were modeled by the students with different experience levels in using Semantic MediaWiki and in modeling processes. Only half of the students have ever used a process modeling tool. After a brief introduction of the basic functionality of our tool, each student was asked to model three assigned processes with it without any hint. The results can be found in our evaluation wiki[8]. The students modeled the processes in different ways. While all students used task and sequence flows, only four students used the gateway elements. Conditions were sometimes expressed in the textual description, in the graphical representation, or in both representations. Additional semantic properties were introduced by half of the students.

   At the end, each student had to fill out a Computer System Usability Questionnaire (CSUQ) [18], by rating 19 statements from 1 to 7 with respect to our tool, where 7 is *strongly agree* and 1 is *strongly disagree*. In addition, the questionnaire was extended with questions about previous process modeling experiences and free text questions about most positive and negative aspects. The results of the CSUQ can be found in Table 1. The overall assessment of the students about the usability of the tool was very positive. Only the quality of the error messages was ranked negative in average. As a consequence, we will improve them in future releases.

   The students criticized that all information entered as wiki text was deleted when they clicked on another task while the wiki text editor was not closed using the *End Edit Mode* button. In addition they recommended that the system should provide duplication of pages within the modeler, in order to enable the faster modeling of similar processes. Some students needed more time in the beginning to get familiar with the wiki syntax. As positive aspects most of the students explicitly mentioned the intuitive usability. It was stated that it was easy to handle, because the user interface was

[7] `http://www.active-project.eu`

[8] The evaluation wiki can be accessed via `http://oryx.f-dengler.de` (Username: ProcessTester – Password: active!).

Table 1. Evaluation Results

Question	N	AVG	DEV
Overall, I am satisfied with how easy it is to use this system	10	5,30	1,35
It was simple to use this system	10	5,50	1,43
I can effectively complete my work using this system	10	5,60	1,28
I am able to complete my work quickly using this system	9	4,67	1,49
I am able to efficiently complete my work using this system	10	5,50	0,81
I feel comfortable using this system	10	5,20	1,47
It was easy to learn to use this system	10	5,40	1,74
I believe I became productive quickly using this system	9	5,56	1,34
The system gives error messages that clearly tell me how to fix problems	6	3,00	1,91
Whenever I make a mistake using the system, I recover easily and quickly	10	5,50	1,36
The information (such as online help, on-screen messages, and other documentation) provided with this system is clear	8	4,63	1,87
It is easy to find the information I needed	9	5,11	1,66
The information provided for the system is easy to understand	9	5,44	1,95
The information is effective in helping me complete the tasks and scenarios	9	5,33	1,63
The organization of information on the system screens is clear	10	5,20	1,60
The interface of this system is pleasant	10	5,90	1,58
I like using the interface of this system	10	5,90	0,94
This system has all the functions and capabilities I expect it to have	6	5,50	1,38
Overall, I am satisfied with this system	10	5,40	1,02

perceived as simple and clear. The user interface was positively mentioned for not being overloaded with unnecessary features.

The results are favorable, and show that the tool can be used intuitively without any previous experience in process modeling. The tool provided the functions needed to complete the tasks. The integration of natural language and graphical elements was widely used. A small number of students used additional features provided by the graphical interface, like coloring graphical elements. The use of such features was not expected by the evaluators, but shows that users can creatively extend the system to cover their needs.

3. To complement our approach, we also implemented the support for the automatic transformation of textual process knowledge articulated in SMW into a graphical process representation displayed in the Oryx component (described in Section 5). In order to evaluate this step, we performed a functionality test by transforming twenty existing textual process descriptions into graphical process representation. We took ten how-to descriptions from WikiHow[9] as examples for processes described on single wiki pages. For processes expressed via predefined semantic properties (i.e. the multi wiki page approach), we used existing processes developed within the ACTIVE project. The transformation of multiple wiki pages to a single process worked as expected. During the transformation of simple how-tos represented as single wiki pages, we detected multiple numbered lists on some how-to pages. As a result, we got two separate process sequences in the graphical representation. We have to check in a further evaluation if we can automatically decide

[9] `http://www.wikihow.org`

if these graphical representations display separated processes or not. The complete results and input data can be found in our demo wiki[10].

7 Related Work

The work presented in this paper is related to the following streams of research: (1) social software for process management, and (2) collaborative process modeling.

There exist other approaches to manage processes with social software [10,20], but none of them supports process knowledge capturing with graphical representations, formal semantic annotations, and natural language. To support workflows, Dello et al. [7] have extended the Makna Semantic Wiki by integrating the workflow engine jBPM. It enables the coordination of interactions within a wiki system, but does not support the collaborative creation of the workflow.

The MoKi enterprise modeling wiki [14], by contrast, allows the collaborative development of processes. However, the tool does not translate the collaboratively created graphical process flow descriptions into machine readable formal semantics, but stores merely a graphical representation. Therefore, SMW queries concerning process semantics, such as *"show me all process activities resulting in an approval activity"* are not possible. A further extension of the Moki enterprise modeling wiki, named BP-MoKi [12], supplements the graphical flow descriptions with formal semantics. However, existing textual process description, previously stored within the wiki, cannot be reused and automatically transformed into graphical process descriptions. Also the semantics of the processes and the other entities in the wiki are in separated stores, and thus cannot be connected natively. Furthermore, the graphical interface of both tools does not allow the user to read and edit the wiki pages within the graphical edit mode. This requires users to switch between graphical and textual mode.

Semantic Wikis can also be used for the management of process model relations. The knowledge about the model relations is captured in a semantic wiki and thus can be searched and reused across different projects and stakeholders [11].

Another approach of using groupware tools for process elicitation is described in [13]. The tool supports collaborative modeling, but does not provide the collaboration functionality of a wiki engine. In addition, the process models have no formal semantics and thus they are not machine processable.

Other BPM tools and Web communities allow collaborative business process modeling, like Activiti[11] and BPMN-Community,[12] both based on Oryx [6], or processWave[13] based on Google Wave. There are also commercial tools, like IBM BPM Blueprint[14] and ARISalign[15]. However, they only focus on the

[10] The demo wiki can be accessed via `http://bpmexample.wikiing.de` (User name: `ProcessTester` – Password: `active!`).

[11] `http://www.activiti.org/`

[12] `http://www.bpmn-community.org/`

[13] `http://www.processwave.org/`

[14] `http://www.lombardisoftware.com/`

[15] `http://www.arisalign.com/`

collaborative development of the process model and require modeling expertise. Novice users cannot model unknown constructs with natural language. Only predefined process properties can be used to further formalize processes. If a property is not included in the process modeling language, it cannot be used. Since the flow structure is only stored in the process diagram and not as semantic descriptions, the search is rather limited compared to our approach.

8 Conclusion

We presented a multi-faceted approach towards the gradual elicitation of knowledge about processes, following the idea of knowledge maturing. We provide a wiki-based implementation of our approach, which combines natural language descriptions, graphical representations, and formal, machine-readable semantics. The mature wiki platform we build upon provides us with numerous basic functionalities which we carefully exploit. This includes user identity management, complete history of the wiki pages, and a well-tested discussion facility. In addition, interlinking between process descriptions and external resources with annotations is supported. In combination with the provided query language, it enables more sophisticated retrieval, browsing and navigation. By storing processes in a machine-processable format (Requirement: **Structured Process Documentation Support**), using a standard exchange formats (RDF), it can easily be integrated into existing semi-automatic process acquisition approaches and enhance their functionality. The tool is available on the Web and can be tested, including the results of the functional evaluation.

We evaluated our tool compared to the requirements that were extracted from current literature and the enterprise case studies that we examined. The functionality was evaluated against a number of real processes taken from case study partners. A usability evaluation further has shown that the tool is superior to alternative approaches for knowledge elicitation from novice users. Natural language or incomplete models can be used to formulate a first idea (Requirement: **Natural Language Support for Novice Users**), which can be further refined and consolidated by other novice users or experts (Requirement: **Collaboration Support**). The rich user interface provides users with means to intuitively model processes graphically with basic (but widely used) process elements, namely *tasks*, *sequence flows*, *parallel* and *exclusive split gateways*. We only used this small sub set of process elements, because with too many elements, the usability of the tool may suffer (Requirement: **Intuitive Graphical Rendering and Editing of Processes**).

In summary we can reasonably expect that the tool implements our process knowledge maturing approach and that this will enable a more holistic approach towards the effective, and efficient, elicitation of reusable process descriptions in the enterprise.

For future work, we will further validate our approach by involving various users collaboratively constructing processes. We will also investigate the appropriate expressivity of the process language for different modeling skills. But due to the open, wiki-based approach, users can already today extend the system with their own elements, creating further expressivity which is exported in RDF, and can then be further processed. Our user evaluation has shown that a small number of power users are able to creatively extend the system to cover their needs, and thus provide a path for the wider usage base to most efficiently explicate the collective process knowledge of a given community.

Acknowledgements. This work was supported by the European Commission through the IST project ACTIVE (ICT-FP7-215040) and by the German Federal Ministry of Education and Research (InterLogGrid, Grant BMBF 01IG09010F).

References

1. van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: a research agenda. Computers in Industry 53(3), 231–244 (2004)
2. Barrett, D.J.: MediaWiki, 1st edn. O'Reilly Media, Inc., Sebastopol (2008)
3. Beckett, D.: RDF/XML syntax specification (Revised). Tech. rep., W3C (2004), `http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/`
4. Bruno, G., Dengler, F., Jennings, B., Khalaf, R., Nurcan, S., Prilla, M., Sarini, M., Schmidt, R., Silva, R.: Key challenges for enabling agile BPM with social software. J. of Software Maintenance and Evolution: Research and Practice 23(4), 297–326 (2011)
5. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: CHI 2004: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems, pp. 175–182. ACM, New York (2004)
6. Decker, G., Overdick, H., Weske, M.: Oryx – An Open Modeling Platform for the BPM Community. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 382–385. Springer, Heidelberg (2008)
7. Dello, K., Nixon, L.J.B., Tolksdorf, R.: Extending the Makna semantic wiki to support workflows. In: SemWiki, CEUR Workshop Proc., vol. 360. CEUR-WS.org (2008)
8. Dengler, F., Lamparter, S., Hefke, M., Abecker, A.: Collaborative Process Development using Semantic MediaWiki. In: Proc. of the 5th Conf. on Professional Knowledge Management, pp. 97–107 (2009)
9. Djordjevic, D., Ghani, R., Fullarton, D.: Process-centric enterprise workspace based on semantic wiki. In: Proc. of the Int. Conf. on Knowledge Management and Information Sharing (KMIS/IC3K) (2010)
10. Erol, S., Granitzer, M., Happ, S., Jantunen, S., Jennings, B., Johannesson, P., Koschmider, A., Nurcan, S., Rossi, D., Schmidt, R.: Combining BPM and social software: contradiction or chance? J. of Software Maintenance and Evolution: Research and Practice 22, 449–476 (2010)
11. Fellmann, M., Thomas, O., Dollmann, T.: Management of model relations using semantic wikis. In: Proc. of the 2010 43rd Hawaii Int. Conf. on System Sciences, HICSS 2010, pp. 1–10. IEEE Computer Society, Washington, DC, USA (2010)
12. Francescomarino, C.D., Ghidini, C., Rospocher, M., Serafini, L., Tonella, P.: A framework for the collaborative specification of semantically annotated business processes. J. of Software Maintenance and Evolution: Research and Practice 23(4), 261–295 (2011)

13. de Freitas, R.M., Borges, M.R., Santoro, F.M., Pino, J.A.: Groupware support for cooperative process elicitation. In: Favela, J., Decouchant, D. (eds.) CRIWG 2003. LNCS, vol. 2806, pp. 232–246. Springer, Heidelberg (2003)
14. Ghidini, C., Rospocher, M., Serafini, L.: Moki: a wiki-based conceptual modeling tool. In: ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, vol. 658, pp. 77–80 (2010)
15. den Hengst, M.: Collaborative modeling of processes: What facilitation support does a group need? In: Proc. of AMCIS 2005, Paper 15 (2005)
16. Koutras, C., Koussouris, S., Kokkinakos, P., Panopoulos, D.: Cockpit public services scenarios. Tech. Rep. D1.1, COCKPIT Project (June 2010)
17. Leuf, B., Cunningham, W.: The Wiki Way: Collaboration and Sharing on the Internet. Addison-Wesley, Reading (2001)
18. Lewis, J.R.: IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. Int. Journal of Human-Computer Interaction 7, 57–78 (1995)
19. Muehlen, M.z., Recker, J.: How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 465–479. Springer, Heidelberg (2008)
20. Nurcan, S., Schmidt, R.: The Third Workshop on Business Process Management and Social Software (BPMS 2010) as part of BPM 2010 (2010)
21. Recker, J., Safrudin, N., Rosemann, M.: How Novices Model Business Processes. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 29–44. Springer, Heidelberg (2010)
22. Russell, N., Hofstede, A.H.M.T., van der Aalst, W., Mulyar, N.: Workflow Control-Flow Patterns: A Revised View. Tech. Rep. BPM Center Report BPM-06-22, BPM-center.org (2006)
23. Santoro, F., Borges, M., Pino, J.: Tell us your process: A group storytelling approach to cooperative process modeling. In: 12th Int. Conf. on Computer Supported Cooperative Work in Design, CSCWD 2008, pp. 29–34. IEEE, Los Alamitos (2008)
24. Schmidt, A., Hinkelmann, K., Ley, T., Lindstaedt, S., Maier, R., Riss, U.: Conceptual Foundations for a Service-oriented Knowledge and Learning Architecture: Supporting Content, Process and Ontology Maturing. In: Pellegrini, T., Auer, S., Tochtermann, K., Schaffert, S. (eds.) Networked Knowledge - Networked Media. SCI, vol. 221, pp. 79–94. Springer, Heidelberg (2009)
25. Schmidt, A., Hinkelmann, K., Lindstaedt, S., Ley, T., Maier, R., Riss, U.: Conceptual foundations for a service-oriented knowledge and learning architecture: Supporting content, process, and ontology maturing. In: 8th Int. Conf. on Knowledge Management (I-KNOW 2008) (2008)
26. Vrandečić, D., Gil, Y., Ratnakar, V.: Want world domination? Win at Risk!: matching to-do items with how-tos from the Web. In: Proc. of the 2011 Int. Conf. on Intelligent User Interfaces, Palo Alto, CA, USA, February 13-16, pp. 323–326 (2011)