Business Process Modeling Support by Depictive and Descriptive Diagrams

Agnes Koschmider[1], Timm Caporale[1], Michael Fellmann[2], Jonas Lehner[1],
Andreas Oberweis[1]

Abstract: The design of a "good" business process model is a time-consuming and error-prone task and requests high training effort from the process modeler. These barriers might be a reason why business processes are often designed with software tools, which were not intentionally developed for this purpose, but are highly familiar for the process modeler (e.g., add-ins for MS Office family) and thus a process model can be quickly designed. As consequence of such a tool choice for process modeling the variety of techniques available for Business Process Management cannot be exploited. To mitigate this situation, we first examine approaches aiming to support business process modeling more intuitively. We then suggest the introduction of an additional layer to business process models with depictive diagrams that are not bounded to a concrete process modeling language or descriptive diagrams using natural language text. We then show how such a layer can be aligned with common process modeling languages and thus provides a seamless integration with more advanced Business Process Management languages and tools. We expect that our approach will fertilize techniques facilitating business process modeling for all types of process modelers including business experts with limited experience of process modeling.

Keywords: Process Modeling, Business Process Model, Natural Language Processing, Visual Variables

1 Introduction

In the literature it is known that unexperienced modelers (e.g. novice modelers or business experts with little training) do not share the same expertise as professional modelers (e.g., business analysts) in terms of applying modeling guidelines and the correct use of the modeling language [KW10]. In more detail, unexperienced modelers tend to forget model elements according to a study from Nielen et al.: *"Concerning error frequencies, activity omissions were considerably higher for novices than for experienced modelers*" [NKM11]. Moreover, according to another study from Wilmont et al., unexperienced modelers have problems in finding the right level of details [WBv10]. From these empirical findings it can be concluded that applying a fully-fledged process modeling language supported by a sophisticated tool is too challenging for unexperienced modelers such as business or domain experts. Although they might have a profound knowledge of the domain that is to be modeled, modeling itself presents

[1] Institute of Applied Informatics and Formal Description Methods, KIT, Karlsruhe, Germany, <firstname.lastname@kit.edu>
[2] Business Information Systems, University of Rostock, Rostock, Germany, michael.fellmann@uni-rostock.de

a barrier for them. It is thus important to reduce this barrier by providing alternative ways of participation while at the same time retaining the richness of fully-fledged process modeling languages for experts. This is still an open issue despite a large body of work suggesting assistance functions for business process modeling [FZM15]. Such approaches surely might help to decrease the effort of process modeling. These assistance tools suggest (similar to an auto-completion function) suitable fragments to complete a currently edited business process model. Definitely, such assistance functions increase the process model quality [KHO11]. However, the assistance is not process modeling agnostic. This means that process modelers still have to be familiar with the process modeling language and technique in order to fully exploit the modeling assistance.

In our research, we hence try to lower the entry barrier to process modeling in a different way. We suggest a lightweight approach to modeling via an on top layer to process models. This layer contains abstract models ("Layer 0") that can be represented both depictive (iconic) or descriptive (symbolic) with the possibility to seamlessly switch between them. This layer should enable a quick and comprehensive view of the underlying process model and in addition should expose basic modelling capabilities. With this layer, we aim to make modelling accessible for a larger audience.

To identify relevant influence factors for the design of such a layer, first, a solid revision of related disciplines emphasizing different modalities of visualizing diagrams is required. This revision is presented Section 2. This section also discusses the range of variables in order to appropriately visualize the diagrams. Based on this discussion, Section 3 suggests two approaches for a graphical and textual visualization. Related approaches are compared in Section 4. The paper ends with a summary in Section 5.

2 Variables to Design Abstract Models

Generally, information can be presented either descriptive or depictive [SB03]. Depictive is related to an *iconic* representation of information where, for instance, graphics are used to describe the context. Descriptive is related to a symbolic representation of information, where natural language text describes the information. While some process model readers prefer textual information, others prefer two-dimensional representations such as graphics [Mo09]. Both modalities are processed differently, which means that different concepts are required for depictive and descriptive representations. According to the dual channel theory [MM03], visual representations are processed in parallel by the human visual system, while textual representations are processed serially by the auditory system [Be83].

When examining the strength of depictive representations, the argumentation of [Ai06] stands out that depictive representation "*can more easily express abstract information and more general negations and disjunctions*". Another strength was observed by [SO95] who argues that "*text permits expression of ambiguity in a way that graphics* *cannot easily accommodate. It is this lack of expressiveness that makes diagrams more effective for solving determinate problems*". It seems to be an agreement that "*visual displays are often said to enhance or "augment" cognition*" [He11].

Despite these strengths of depictive diagrams, descriptive representations are also advantageous. Particularly, descriptive (textual) diagrams have the advantage that no prior training effort is required in order to understand and use the symbols (letters, words). Furthermore, in new directions (e.g., mobile process modeling) textual diagrams, which suit to be created on the go, can be used as an intermediate from which the graphical diagram are generated automatically [Ke14].

Obviously, information can be presented in both modalities combining graphics with textual description. Presenting information in multiple modalities is regarded as being useful to learners who actively process such information [Ai06]. On the other hand, Kalyuga [Ka11] observed that the human working memory is very limited when handling new information because initially no mechanism is available that coordinates novel information. Due to the restricted capability of the working memory Kalyuga advocates to separate channels for dealing with auditory (descriptive) and visual material (depictive).

To sum up the discussion, both modalities of representation have their strengths. To exploit them, the abstract layer suggested in this paper, offers both modalities in order to support particularly business experts with limited experience of process modeling. The diagrams created on this abstract layer can be considered as abstraction of the underlying process models, meaning that fine-grained concepts are abstracted to related concepts on a higher level. Both representations are aligned in order to allow a seamlessly switch between them and they allow a navigation to their subsequent layers in order to support a seamless integration with techniques and tools of Business Process Management. Fig. 1 shows the placement of the new layer. We call it "Layer 0" since it precedes the current starting point of process modeling on e.g., "Layer 1".

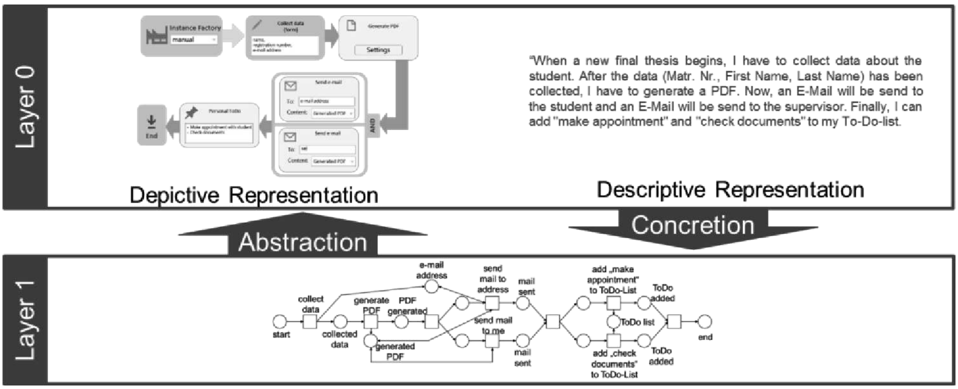

Fig. 1: Two modalities of the abstraction layer ("Layer 0")

The new layer, which allows "textual abstraction" and "graphical abstraction", has several advantages:

- From the viewer's perspective: Both representations should allow the process model's viewer to get a quick and comprehensive view of the underlying process model. If the viewer is further interested in the fine-grained view of the process model he/she can navigate through the process model hierarchy.
- From the creator's (i.e., process modeler's) perspective: the concepts of this layer abstract from common process modeling languages, and thus, we expect, that the creation of process models even for inexperienced persons is easier.

In the following we discuss variables how to best design both modalities of representation.

2.1 Designing Depictive Diagrams

The design of a depictive diagram can be described based on the visual variables by J. Bertin [Be83]. These visual variables have been applied to process models by [Ko15] and are summarized in this section. Visual (or graphical) presentation is categorized in planar variables (addresses the X, Y location) and retinal variables (shape, size, color, brightness, orientation, texture). Some of these variables are detected in parallel (color and texture). Shape, for instance, is detected in a less efficient scanning [TS86]. Thus, scanning of shape is affected when combining it with color. This means that business process models have to be designed in a way that users can recognize the fundamental elements of the model with minimal cognitive effort. Each of these variables can be used singularly or in combination.

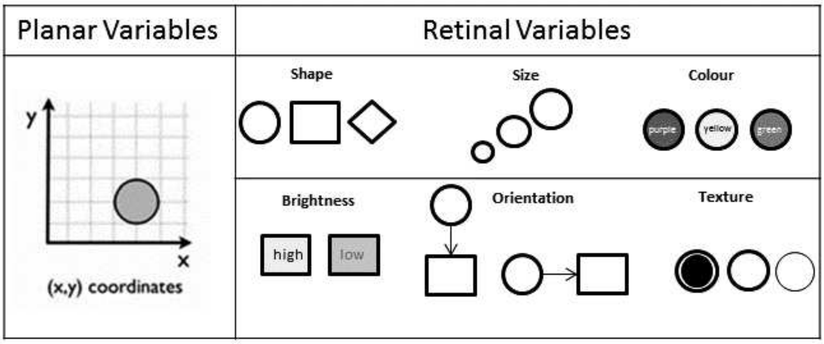

Fig. 2: Visual variables

Shape. Different information can be expressed by different shapes. A varying number of visual variables of graphical elements makes the elements easier to identify [Mo09]. To avoid confusion, it is recommended to use common (prominent) and particularly distinct geometrical shapes (circle, triangle, square, diamond). Following such recommendations of symbol choice increases memory of visual aspects [Fi12]. Shapes are appropriate for the representation of information and the shape choice should be well considered.

Size. Size of a process element must be in relationship to the total number of elements of the process model, the length of the element label and the space of the modeling workbench. In this context it should be considered that [KRD12] found out, when using the preferred style of granularity (flattened process models with no refinements versus modularly built process models) then no negative effects were identified on the performance in making sense of such a process model. This means that “large” process models might be understandable if such a process model corresponds to the preferred style of granularity of the user.

Color. Color is a powerful and effective visual variable because it is detected in parallel [TS86]. Differences in color are perceived faster than differences in shape. Generally, color facilitates information processing [Lo93], when being used effectively. Too many different colors however can impair communication [LGH14] and do not act as effective cognitive aids in problem solving When using color as visual variable in order to represent the context on layer 0, it might be rational to limit the number of colors to the use of six colors for symbols since it is found most efficient with respect to readability [Pi08].

Brightness[3]. Few empirical studies exist, which show that these two visual variables improve the readability of graphs or business process models respectively. Identical assumptions are applied for hue and texture as for color (minimize the number of used colors; consider color usability). The empirical study of [KKR11] that subsumes hue under the color aesthetics indicates a stronger preference for color (hue) over brightness for the purpose to visualize changes in business process models.

Orientation. The constructs should be shown in a way that an orientation of the diagram is evolved by the user (mixing of orientation should be avoided). An initial investigation on process model orientation indicates a benefit with respect to readability for a left-to-right flow direction [FS14].

2.2 Designing Descriptive Diagrams

Since no significant training effort is required to “create” a descriptive representation, this representation can easily be used. Moreover, the creation of descriptive diagrams should be a common feature of BPM systems (which is mostly not the case). The creation of diagrams based on this kind of representation would allow process modelers to create “good” business processes in an appropriate way on their own, without having

[3] Texture and brightness are not elaborated separately in our context. Brightness and texture (hue) are considered as components of color aesthetic and thus identical assumptions can be applied for hue and texture as for color.

deep knowledge about process modeling languages. For instance, an autocompletion function could be integrated in order to provide lexical templates to be selected for the creation of textual process descriptions. Subsequently, a grammar or syntax has to be defined for the diagram. However, a disadvantage of natural language usage for communication (as process models intent to) is ambiguity. Descriptive or textual diagrams are described using natural language, which is called to cause ambiguity.

When using natural language expressions for diagrams an efficient parsing (decomposition of sentences) should be supported. Generally, a sentence can be decomposed according to the phrase structure grammar [LC57]) or dependency grammar. Dependency refers to the notion that relationships between linguistic units (e.g., words) are directly linked to each other. Grammatical relationships are preserves between linguistic units. The phrase structure grammar also decomposes sentences to linguistic units using a phrase structure tree, which is a recursive decomposition of the whole word sentence into smaller sentences, down to one-word unit without preserving the dependents between the linguistic units [Lo98]. Comparisons between both types of grammar confirm an efficient parsing for dependency grammars.

To sum up our considerations, both types of diagrams have advantages and allow business experts with a limited experience of modeling to create diagrams. Generally, a switch between both modalities should be offered in order to lower the entry barrier to modeling for unexperienced modelers. Based on these considerations, the next section presents two work in progress approaches for the design of depictive and descriptive diagrams for an on- top layer (“Layer 0”).

3 Approaches for Descriptive and Depictive Diagrams

3.1 Generation of depictive diagrams

The design of a depictive diagram for a business process is based on the guidelines discussed in Section 2.1. For this purpose, we use a technique stemming from design thinking (see e.g., [LW11]). In so-called “Tangible Business Process Modeling” plastic elements, which correspond to BPMN iconography, are used to model business processes through play. Particularly, this approach is suitable for process modelers with limited modeling experiences. After the creation of the tangible process model, the process model has to be enriched with additional information in order to make it automatically processible and executable.

As an example we describe the registration of a thesis at a university from the supervisor’s perspective, who first collects data from the student, then generates several documents (e.g., registration form), sends them to the student’s and his own mailbox, and adds two tasks to the supervisor’s personal task list. The process model is shown in Fig. 3. The user selects the needed process activities from the palette on the right hand side of the tool. The look and feel of the interface is designed closely to the Microsoft Office products family. For instance, the ribbon bar, the fonts and names of the menu items are imitated from MS Office product family. Thus, the user should feel familiar when working with the tool. Also his/her familiarity with the MS Office product family should increase the tool acceptance.

This visualization approach uses the visual variables discussed in Section 2.1 particularly (1) color, (2) shape and (3) symbols are combined allowing an efficient scanning of the process model also particularly by unexperienced users.

Color: The starting point is dyed green ("Instance factory") and the endpoint is red, which allows the process modeler to quickly identify the starting and end point and thus the range of the diagram. Nodes, which require the user's interaction, have blue color. All other nodes (workflow activities) are grey. The contrast between grey and the three used colors is high. Altogether, the number of used colors is well-balanced.

Symbols: Each node has a little symbol in the upper left corner, which represents the activity which is performed in this step. The usage of symbol and text allows the user to quickly select the needed construct.

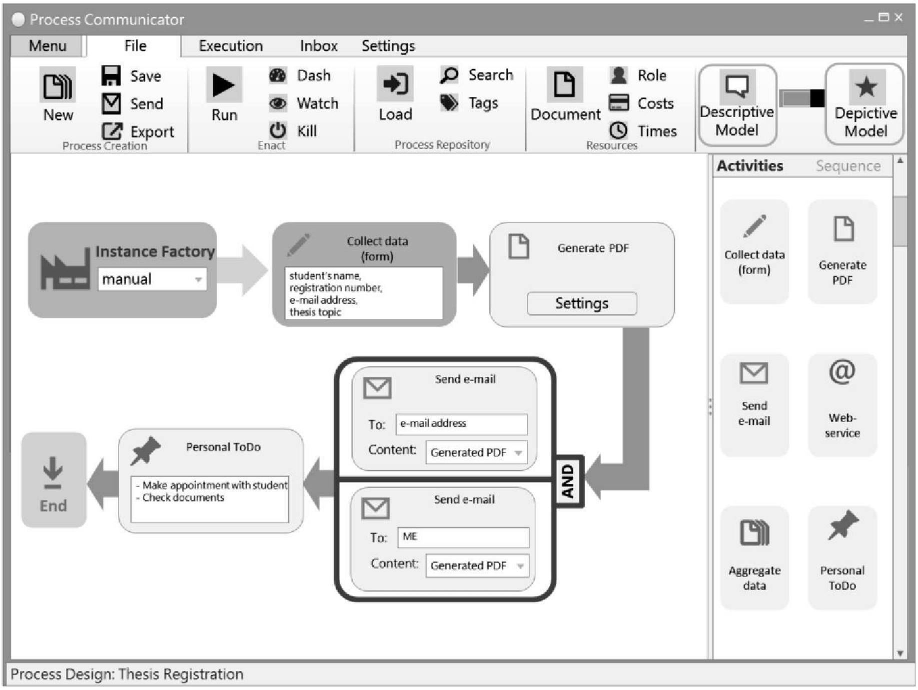

Fig. 3: Approach for a depictive representation ("Layer 0") in a workflow management system [Le15]

The process model's orientation is evolving as the model grows. That is, in the beginning new elements are inserted from left to right. If one line is full, a line wrap is inserted automatically and new elements are added from right to left. The size of the symbols is changed automatically related to the number of nodes used within the workspace. Brightness is used to highlight special characteristics ("AND" node).

To allow navigation to the subsequent process model a transformation from the depictive diagram to a Petri net is supported as shown in Fig. 4.

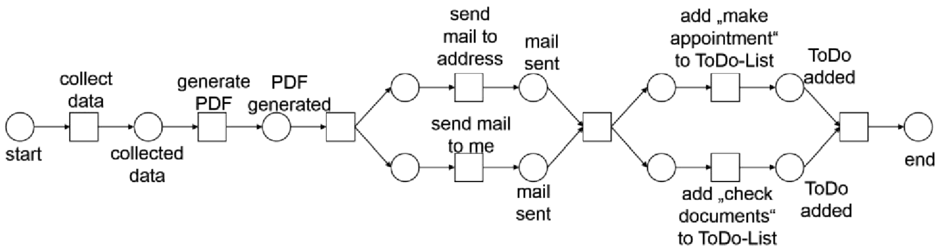

Fig. 4: Petri net resulting from a transformation of the depictive diagram showing the registration process from the supervisor's point of view.

A sequence of activities on the Layer 0 is also translated in a sequence according to the workflow control-flow patterns. The "AND" node is translated into a Parallel Split and Synchronization pattern. Analogously a "XOR" node used in the depictive diagram on Layer 0 would result in an Alternative and Simple Merge pattern.

This automatically generated process model can be further enriched by advanced process modeling by adding additional information, e.g. data objects, which could result in a more detailed Petri net as shown in Fig. 5.

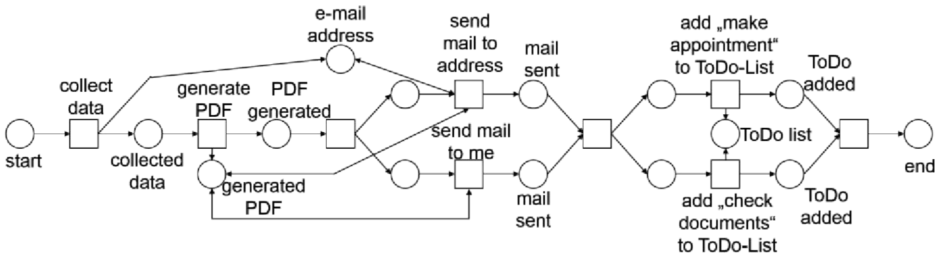

Fig. 5: Process model enriched with additional information by a modeling expert

The suggested visualization approach has been developed according to the design options and guidelines introduced in the Section 2.1. The next section summarizes an approach how to generate graphics from text within the same tool.

3.2 Generation of descriptive diagrams

Descriptive approaches aim to process natural language text. The approach presented in this section is based on a formal model allowing a bidirectional transformation from text to graphics by [Ke14]. This model has been further refined by the introduction of *sentence templates* supporting an efficient decomposition of sentences from natural language text [Ca15]. Assume that Petri nets should be generated from natural language text, the underlying concept can be translated into extended Backus-Naur Form as follows:

```
1:  start: sentences*;
2:  sentences: placeStart | transitionStart;
3:  placeStart: prefix-pl placesList postfix-pl
    prefix-tr transitionList '. ';
4:  prefix-pl: 'If ' | 'After ' | 'When ' |
    'As soon as ' | 'In case of ';
5:  postfix-pl: ' happened, ' | ' was typed in, ' |
    ' came in, ' | ' is valid, ' | ' is unvalid,' | ', ';
6:  prefix-tr: ' I can ' | ' I have to ' |
    ' the system must ' | ' the activity ' | ' then ';
7:  placesList: place | ' either ' place ' or '
    furtherplaces | place ' and ' furtherPlaces;
8:  furtherPlaces: place | place  ' or ' furtherPlaces |
    place ' and ' furtherPlaces;
9:  transitionList: transition | ' either ' transition
    ' or ' furtherTransitions | transition ' and '
    furtherTransitions;
10: furtherTransitions: transition | transition
    ' or ' furtherTransitions | transition  ' and '
    furtherTransitions;
11: transitionStart: 'Now, ' transitionList '. ' |
    prefix-tr transitionList '. ';
12: place: content;
13: transition: content;
14: content: STRING+ (' ' | STRING )*;
15: STRING: (~(' '|'.'))+;
```

Fig. 6 shows an exemplary User Interface for a descriptive approach. To use this kind of “modeling” does not require any knowledge of process modeling. Instead the natural language techniques are used to transform the natural language to graphical elements. The sentences can be either typed in manually or they can be recorded and processed by a voice-to-text recognition tool. After the insertion of text, process pattern recognition takes place and the recognized patterns are visualized and displayed immediately. The natural language is exploited in two ways. A bidirectional link between the graphical process model and the textual process description allows checking the correspondence between the spoken or typed text and the graphical process model at any time. Additionally, we have developed a modeling assistant. Support is available through an automatic selection of natural language templates, which assist in the formulation of sentences of the underlying grammar. The syntax of the templates depends on the modeling language syntax.

The natural language text is inserted on the left hand side, while the corresponding patterns are visualized on the right hand side.

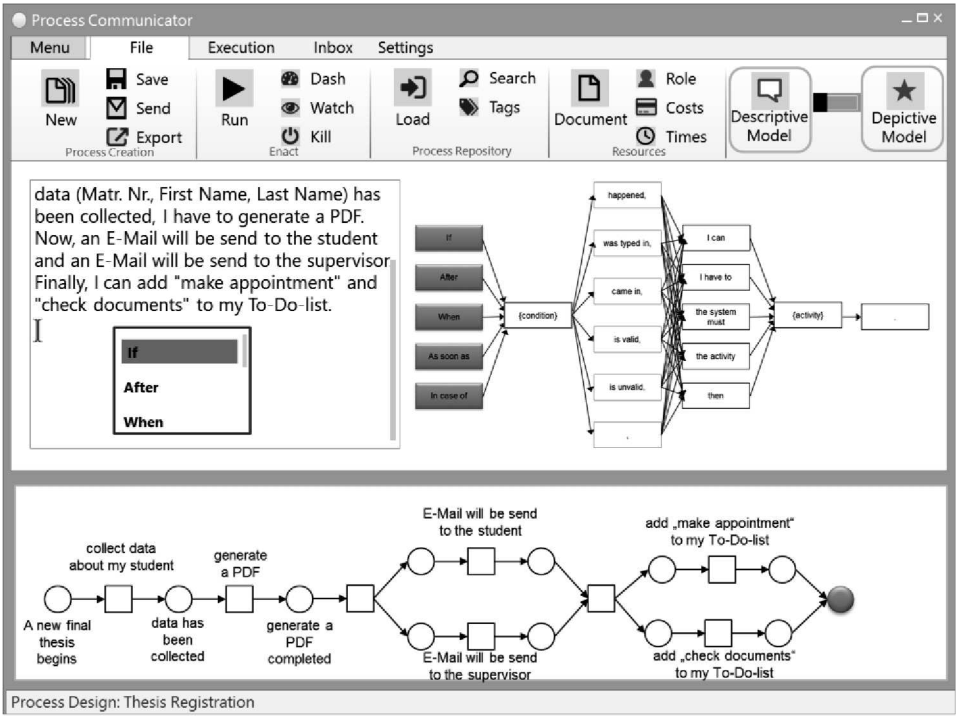

Fig. 6: Approach for a textual abstraction

The templates on the right are connected to a specific modeling pattern and automatically show alternative formulation variants. A sentence is composed by lining up the possible elements, which is illustrated by the connecting arrows. Yellow elements mark placeholders for conditions and blue elements placeholders for activities. The user will automatically been shown up examples when the text input comes either to the text relevant placeholder or the user clicks on the placeholder by mouse. Using our tool, it is possible to create traditional models (Level 1) and display an abstract depictive or descriptive model based on the model. Moreover, it is possible to switch between the depictive and descriptive view in a fast and seamless way.

4 Related Work

Several related scientific works as well as developments from the software industry address the barrier for knowledge externalization.

In regard to modeling methods, much research has been conducted on how novice modelers are constructing process models. In this area, it is known from empirical studies that novice modelers struggle to create "good" process models since they tend to forget important model elements [NKM11] or have problems in finding the right level of abstraction [WBv10] These observations support our goal of creating a layer for simplified modeling. Also, empirical insights suggest that the combination of abstract graphical symbols (depictive) in conjunction with describing text (descriptive) improves model comprehension for unexperienced modelers [RSR12] which is in line with our ambitions to combine both.

In order to provide abstract graphical models, research has devised techniques to automate process model abstraction [PSW15]. Moreover, in order to switch between models and text, research in the intersection of linguistics and BPM has put forth techniques to generate process models from text [FMP] and vice versa [LMP12]. These approaches are designed in order to transform texts or models, i.e. to be applied *before* or *after* modeling, while our approach is intended to support modeling itself and hence to be applied *during* modeling. In regard to similar modeling approaches, Process Chain Diagrams (PCD, in German "Vorgangskettendiagramm") [Sc13] already intended to provide a high-level overview layer over a set of more detailed process models that may be linked to the PCD. However, in contrast to our approach, this layer has to be created and updated manually. Another approach hence is to omit the detailed layer and exclusively focus on models that are somewhere in the middle of the granularity continuum ranging from detailed task-oriented models to coarse-grained PCDs. An approach in this direction PICTURE [BPR07]. It offers a lightweight domain-specific language providing a vocabulary and set of symbols to efficiently capture the processes of public administrations. In contrast to this approach, our aim is a generic approach to facilitate the access to Business Process Modeling that is not bound to a specific domain or modeling language. Another approach is the Guarded Process Spaces (GPS) approach [RDR12]. It is applied in the domain of hospitals where process management is important. With GPS, business users can model executable process templates and moreover flexibly adapt running process instances. Both are accomplished using a "navigation paradigm". This means that the end user is guided in modeling as well as in performing ad-hoc deviations during runtime. In contrast to our approach, this approach is also domain specific. It moreover mixes modeling with execution which is beyond the scope of our work.

In respect to modeling tools, related approaches focus on alternative process model presentations that are easier to understand than e.g. fully-fledged BPMN process models. For example, the *Signavio Process Editor* (cf. www.signavio.com/products/process-editor) provides a mechanism "Quick Model" that allows basic process modeling based on a spreadsheet-like working platform. With this feature, the product aims to involve all participants in the process design, even those that are not capable of process modeling. The mechanism is based on filling out tables with start- and end-events. In addition, incoming and outgoing documents as well as different roles are assigned to the respective process steps. The tool then generates a process model in BPMN 2.0 notation. Another tool *Blueworks Live* from IBM (cf. www.blueworkslive.com) provides a similar feature. It moreover allows switching forth and back between the lightweight table-based process presentation and the more traditional BPMN-based process model representation.

5 Conclusion and Future Work

This paper first elicits and discusses variables of how to best design diagrams consisting of graphical or textual elements on top of business process models. It then suggests concrete approaches to the design and implementation of such a layer in terms of the necessary functionality and required user interface. These approaches may pave the way for the detailed specification of requirements and elicitation of further design options and choices. These, in turn, can ultimately result in the development of an explanatory design theory [BP10] for on the top layer modeling support systems.

One direction for the future is the complete implementation and user evaluation of both approaches for (abstract) descriptive and depictive design of process models.

References

[Ai06] Ainsworth, S.: DeFT: A conceptual framework for considering learning with multiple representations. In: Learning and Instruction, 2006, 16; p. 183–198.

[Be83] Bertin, J.: Semiology of graphics: diagrams, networks, maps, 1983.

[BP10] Baskerville, R.; Pries-Heje, J.: Explanatory design theory. In: Business & Information Systems Engineering, 2010, 2; p. 271–282.

[BPR07] Becker, J.; Pfeiffer, D.; Räckers, M.: Domain specific process modelling in public administrations-the PICTURE-approach. In: Electronic Government. Springer, 2007; p. 68–79.

[Ca15] Caporale, T.: A Method for Modeling and Analyzing Business Processes for Knowledge Carriers. In: Proceedings of the 7th Central European Workshop on Services and their Composition, Jena, 2015; p. 18–21.

[Fi12] Figl, K.: Symbol choice and memory of visual models. In: Visual Languages and Human-Centric Computing (VL/HCC), 2012 IEEE Symposium on, 2012; p. 97–100.

[FMP] Friedrich, F.; Mendling, J.; Puhlmann, F.: Process model generation from natural language text. In: Advanced Information Systems Engineering, 2011; p. 482–496.

[FS14] Figl, K.; Strembeck, M.: On the Importance of Flow Direction in Business Process Models, 2014.

[FZM15] Fellmann, M. et al.: Requirements Catalog for Business Process Modeling Recommender Systems, 2015.

[He11] Hegarty, M.: The cognitive science of Visual-Spatial displays: Implications for design. In: Topics in cognitive science, 2011, 3; p. 446–474.

[Ka11] Kalyuga, S.: Cognitive load theory: How many types of load does it really need? In: Educational Psychology Review, 2011, 23; p. 1–19.

[Ke14] Keuter, B.: Bidirektionale Abbildung zwischen Geschäftsprozessmodellen und IT-Kommunikationssystemen. KIT Scientific Publishing, 2014.

[KHO11] Koschmider, A.; Hornung, T.; Oberweis, A.: Recommendation-based editor for business process modeling. In: Data & Knowledge Engineering, 2011, 70; p. 483–503.

[KKR11] Kabicher, S.; Kriglstein, S.; Rinderle-Ma, S.: Visual change tracking for business process models. In: Conceptual Modeling-ER 2011. Springer, 2011; p. 504–513.

[Ko15] Koschmider, A.: Identifying Impacts on the Quality of Business Process Models: A Bottom-Up Approach. Technical Report. KIT, 2015.

[KRD12] Koschmider, A.; Reijers, H. A.; Dijman, R.: Empirical Support for the Usefulness of Personalized Process Model Views. In: Multikonferenz Wirtschaftsinformatik, 2012.

[KW10] Karagiannis, D.; Woitsch, R.: Knowledge engineering in business process management. In: Handbook on Business Process Management 2. Springer, 2010; p. 463–485.

[LC57] Lees, R. B.; Chomsky, N.: Syntactic Structures. In: Language, 1957, 33; p. 375–408.

[Le15] Lehner, J.: Personal BPM–Bringing the Power of Business Process Management to the User. In: Proceedings of the 7th Central European Workshop on Services and their Composition, Jena, 2015; p. 22–25.

[LGH14] Li, L.; Grundy, J.; Hosking, J.: A visual language and environment for enterprise system modelling and automation. In: Journal of Visual Languages & Computing, 2014, 25; p. 253–277.

[LMP12] Leopold, H.; Mendling, J.; Polyvyanyy, A.: Generating natural language texts from business process models. In: Advanced Information Systems Engineering, 2012; p. 64–79.

[Lo93] Lohse, G. L.: A cognitive model for understanding graphical perception. In: Human-Computer Interaction, 1993, 8; p. 353–388.

[Lo98] Lombardo, V.: A computational model of recovery. In: Reanalysis in Sentence Processing. Springer, 1998; p. 287–325.

[LW11] Luebbe, A.; Weske, M.: Bringing design thinking to business process modeling. In: Design Thinking. Springer, 2011; p. 181–195.

[MM03] Mayer, R. E.; Moreno, R.: Nine ways to reduce cognitive load in multimedia learning. In: Educational psychologist, 2003, 38; p. 43–52.

[Mo09] Moody, D. L.: The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. In: Software Engineering, IEEE Transactions on, 2009, 35; p. 756–779.

[NKM11] Nielen, A. et al.: An empirical analysis of human performance and error in process model development. In: Conceptual Modeling-ER 2011. Springer, 2011; p. 514–523.

[Pi08] Pietersma, D.: Symbol Type and Colour in Graphs. In: Pharmaceutical Users Software Exchange, 2008; p. 1–7.

[PSW15] Polyvyanyy, A.; Smirnov, S.; Weske, M.: Business process model abstraction. In: Handbook on Business Process Management 1. Springer, 2015; p. 147–165.

[RDR12] Reuter, C. et al.: Guarded process spaces (GPS): A navigation system towards creation and dynamic change of healthcare processes from the end-user's perspective. In: Business Process Management Workshops, 2012; p. 237–248.

[RSR12] Recker, J.; Safrudin, N.; Rosemann, M.: How novices design business processes. In: Information Systems, 2012, 37; p. 557–573.

[SB03] Schnotz, W.; Bannert, M.: Construction and interference in learning from multiple representation. In: Learning and Instruction, 2003, 13; p. 141–156.

[Sc13] Scheer, A.-W.: ARIS—Modellierungsmethoden, Metamodelle, Anwendungen. Springer-Verlag, 2013.

[SO95] Stenning, K.; Oberlander, J.: A cognitive theory of graphical and linguistic reasoning: logic and implementation. In: Cognitive science, 1995, 19; p. 97–140.

[TS86] Treisman, A.; Souther, J.: Illusory words: The roles of attention and of top-down constraints in conjoining letters to form words. In: Journal of Experimental Psychology: Human Perception and Performance, 1986, 12; p. 3.

[WBv10] Wilmont, I. et al.: Exploring intuitive modelling behaviour. In: Enterprise, Business-Process and Information Systems Modeling. Springer, 2010; p. 301–313.