User Centered Development of Agent-based Business Process Models and Notations

Robert Singer
Department of Computer Sciences
FH JOANNEUM – University of Applied Sciences
Graz, Austria
robert.singer@fh-joanneum.at

Abstract—We discuss questions about user centric development of business process modeling notations. In the center of our research there is a fully featured multi-enterprise business process platform (ME-BPP) based on the concepts of agent-based business processes, which builds on the formal foundations of the subject-oriented business process management methodology (S-BPM). The platform is implemented based on cloud technology using commercial services. Additionally we developed a "block modeling" technique to find a semantically transparent modeling notation which can be used by novice users to model subject-oriented business process (S-BPM) models. As this is ongoing research there are still serious open questions. But, the presented approach breaks with some of the rules of typical process modeling notations and hopefully stimulates innovation. Additionally we want to continue our research towards the enhancement of our modeling approach towards a user centric "syntax and semantic free" modeling technique to develop user and domain specific modeling notations.

Keywords—BPM, S-BPM, Agent, Modeling, Syntax, Ontology, Notation

I. INTRODUCTION

Latest developments in business and technology driven new business models foster more than ever the need for mature business process management (BPM) methodologies and corresponding supporting technologies for distributed business processes, so called choreographies. Business processes cannot be seen as isolated workflows for administrative purposes, but as a mean to coordinate a value system with supply chain partners. Communication is the very nature of a business process choreography – or, in other words, any choreography is a set of structured communication patterns. That means a choreography defines how work is done, taking into account all involved organizations. Distributed execution of a business process means that every process participant may use their own process execution engine. The overall process is then executed by interconnecting multiple engines. The engines may even run on a mobile device.

This demand is reflected in new developments in the domain of BPM, such as BPM Platform as a Service (bpmPaaS), multi-enterprise Business Process Platform (ME-BPP), Cloud BPM, and Social BPM. The term bpmPaaS can be defined [1] as “the delivery of BPM platform capabilities as a cloud service by a service provider”. A ME-BPP is defined [1] as “high-level conceptual model of a multistakeholder environment, where multi-enterprise applications are operated. multi-enterprise applications are those that are purposely built to support the unique requirements for business processes that span more than one business entity or organization. They replace multiple business applications integrated in serial fashion”.

For a distributed execution of a process, two important prerequisites are needed: a suitable process modeling technique and a flexible communication platform [2]. As elaborated in the following sections, we have chosen the agent based approach to model a distributed system. To be more specific, we build on the Subject-oriented BPM methodology, as defined in [3].

To implement a communication platform, we need an architecture, which includes a graphical business process and/or rule modeling capability, a process registry/repository to handle the modeling metadata, a process execution and either a state management engine or a rule engine as minimal requirements. To realize a bpmPaaS and/or ME-BPP system a cloud infrastructure is needed to model and execute processes which span across more than one business entity or organization.

A. Agent-based BPM (AB-BPM)

As already discussed, for example recently by [4] or [5], traditional BPM and its supporting technology frameworks (we mean business process management – or better workflow management – systems (BPMS, WfMS) as supporting layer for process enactment) have some conceptual and practical limitations. Business processes are more than algorithmic workflows (input, black box, output); they often may have deep social aspects, as long as human participants are involved. Therefore, a new view on business processes recently has been promoted under the term Agent-based BPM (AB-BPM) [6] or Subject-oriented BPM (S-BPM) [7].

There is already a long history of the idea of interacting agents. The application of the agent concept into the domain of BPM has emerged from the domain of distributed software [8] by Albert Fleischmann, who developed the Subject-oriented BPM (S-BPM) methodology in the early 2000s based on his PASS [9] language. All language constructs of PASS can be transformed down to pure CCS [10]. The S-BPM methodology enhances the process algebra languages by graphical representations and adds some technical feature definitions.

1Parallel Activities Specification Scheme
2Calculus of Communicating Systems
B. Distributed BPM

Any collaboration contains more than one subject, so it is per definition a multi agent system, which is a subclass of concurrent systems [11] – which is an important fact as it has consequences for a possible technical implementation.

The problem of synchronizing multiple processes is not trivial and has been widely studied through the 1970s and 1980s [12]. To understand the problem [11], it is helpful to first consider the way that communication is treated in the object-oriented programming paradigm, that is, communication as a method invocation. The crucial point is, that an object does not have control over the execution of its own public methods – any other object can execute the object’s public methods whenever they want.

C. Contribution

Recently we presented an implementation of a ME-BPP based on our Structured Information and Communication Technology (StrICT) framework, which is based on the S-BPM methodology; the technical aspect have been discussed in [13] and [14].

In this paper we will focus on the modeling aspects of business processes in general with a concrete application on modeling of agent-based respectively subject-oriented BPM. Further on we will discuss the concept of a “semantic and syntax free” modeling approach we developed as a generalization of our findings; this is ongoing work, including the development of a prototype implementation using a touch sensitive interface to create models and to create user based modeling languages. Finally we will discuss further possible research directions.

II. HOW TO MODEL

In this chapter we will work out requirements for and pitfalls of modeling languages. This discussion will show, that it is not sufficient to precisely define the semantics, but also the syntax of a modeling language. We will also argue, that the predominantly discussed BPMN 2.0 language does not fulfill these requirements to create “good” models.

A. The Notion of Model

Wand and Weber [15] start their seminal work “An Ontological Model of an Information System” with the following – and still valid – remarks:

“The computer science (CS) and information systems (IS) fields are replete with fundamental concepts that are poorly defined.”

A similar discussion about this topic – for example – can be found in “The FRISCO Report” [16]. Based on these and other discussions, we think it is very fruitful to start the discussion with the notion of ontology. Doing this we can try to build a better understanding of a systemic view on organizations; this leads us to a better understanding of the notion of model in general and of process models in particularly.

Business processes offer a dynamic view on organizations as they describe the states and state transitions of a system or the corresponding world. This also defines the ontological model of a system, as elaborated by Dietz [17]: “The ontological model of a world consists of the specification of its state space and its transition space”.

If we now want to model a world, we have to understand the meaning triangle from semiotics as shown in Figure 1.

![Figure 1: The meaning triangle. [17]](image)

A sign is an object that is used as a representation of something else and is used to communicate the concepts in our mind. A well-known class of signs are the symbolic signs (structures put into physical substrates). An object is an observable and identifiable individual thing; objects are concrete or abstract. A concept is a subjective individual thing (a thought in our mind). Denotation and denotation are relevant when we want to communicate.

A precise formal definition of the construction of a system can be found for example in [18], which can be described as: something is a system if it has the following properties:

- **Composition**: a set of elements of some category (physical, social…)
- **Environment**: a set of elements of the same category; the composition and the environment are disjoint
- **Structure**: a set of influence bonds among the elements in the composition, and between them and the elements in the environment

Dietz further on adds the notion of production [17]:

- **Production**: the elements in the composition produce things (goods or services) that are delivered to the elements in the environment. Now, an organization or company is a collection of socially linked human beings; an overview about the different concepts mentioned is depicted in Figure 2.

Following the argumentation of Dietz [17], three gross categories of systems can be distinguished: concrete systems, symbolic systems, and conceptual systems as depicted in Figure 3.

![Figure 3](image)

The conceptual model of a concrete system is called a conceptualization; for example, a business process model is a conceptualization of the business processes of an organization or firm. A concrete model of a conceptual system is called an implementation; for example an enacted business process is
textual representations are processed serially by the language system [19]. Only diagrammatic presentations are able to show (complex) relations at once.

The anatomy of a visual notation is worked out very clearly by [21]:

A visual notation (or visual language, graphical notation, diagramming notation) consists of a set of graphical symbols (visual vocabulary), a set of compositional rules (visual grammar) and definitions of the meaning of each symbol (visual semantics). The visual vocabulary and visual grammar together form the visual (or concrete) syntax. Graphical symbols are used to symbolize (perceptually represent) semantic constructs, typically defined by a metamodel. The meanings of graphical symbols are defined by mapping them to the constructs they represent. A valid expression in a visual notation is called a visual sentence or diagram. Diagrams are composed of symbol instances (tokens), arranged according to the rules of the visual grammar.

But, just presenting information in a graphical form does not guarantee that it will be worth a thousand of words [22]. Most effort is spent on designing semantics, with visual syntax often an afterthought [21]. For example, UML does not provide design rationale for any of its graphical conventions [21].

A widely accepted way to evaluate notations is ontological analysis; the most used ontology seems to be the Bunge-Wand-Weber (BWW) ontology [15]. Ontological analysis involves a two-way mapping between a modeling notation and an ontology. The interpretation mapping describes the mapping from the notation to the ontology; the representation mapping describes the inverse mapping [23] as depicted in Figure 4.

If construct deficits exist, the notation is ontologically incomplete; if any of the other three anomalies exist, it is ontologically unclear. The BWW ontology predicts that ontologically clear and complete notations will be more effective. As elaborated in [21], ontological analysis focuses on content rather than form; if two notations have the same semantics but different syntax, ontological analysis cannot distinguish between them. Moody [21] has developed a promising foundation to analyze the syntactic aspects of notations in a similar stringent way based on scientific foundations.
There is a set of principles based on two core concepts \[21\]. we will summarize here as input for further discussions. The developed theory is a so called Type IV theory \[24\]: a theory for explaining and predicting (how and why). At the top level there is the well accepted theory of communication \[25\] and its application to the domain of visual notations:

\[
\text{..., a diagram creator (sender) encodes information (message) in the form of a diagram (signal) and the diagram user (receiver) decodes this signal. The diagram is encoded using a visual notation (code), which defines a set of conventions that both sender and receiver understand. The medium (channel) is the physical form in which the diagram is presented (e.g., paper, whiteboard, and computer screen). Noise represents random variation in the signal which can interfere with communication. The effectiveness of communication is measured by the match between the intended message and the received message (information transmitted).}
\]

Bertin \[19\] identified eight visual variables that can be used to graphically encode information as depicted in Figure 5. The decoding side is based on the human decoding processes, which can be divided in two phases: perceptual processing (seeing) and cognitive processing (understanding). As the perceptual processing system is much faster, it is more effective to move as much of the decoding work from the cognitive to it.

![Visual Variables](image)

**Fig. 5: Visual variables.** \[19\]; adapted from \[21\]

Now, based on these theories and empirical evidence Moody has developed a prescriptive theory for visual notations \[21\], which is formulated as nine principles for designing cognitively effective visual notations, summarized as follows:

- **Semiotic clarity**: there should be a 1:1 correspondence between semantic constructs and graphical symbols
- **Perceptual discriminability**: different symbols should be clearly distinguishable from each other
- **Semantic transparency**: use visual representations whose appearance suggests their meaning
- **Complexity management**: include explicit mechanisms for dealing with complexity
- **Cognitive integration**: include explicit mechanisms to support integration of information from different diagrams
- **Visual expressiveness**: use the full range and capacities of visual variables
- **Dual coding**: use text to complement graphics
- **Graphic economy**: the number of different graphical symbols should be cognitively manageable
- **Cognitive fit**: use different visual dialects for different tasks and audience

The method of ontological analysis and the set of principles for designing cognitive effective visual notations, together with the understanding of the notion of model and semiotics assembles a full set of building blocks for a coherent and solid foundation for business process modeling notations. Finally, combining it with a corresponding formal model for business process execution leads to a full theory of business process.

**C. Business Process Modeling**

As there is (yet) no coherent and **general accepted** theory of business processes and business process management (BPM) \[5\] \[26\] \[4\], any way to define process models is the right one; for this purpose domain specific languages (e.g. notations such as BPMN or UML) are defined and it cannot be denied that most of them are rooted in the information systems domain. This is a consequence of the fact that information systems engineers need formally defined models without any **semantic ambiguity**. Additionally, modeling is typically conducted by experts, i.e. business analysts and/or requirements engineers. But studies \[27\] show that end users understand such expert models very poorly. One of the reasons for this is that it is hard for experts to think like novices, a phenomena called the **curse of knowledge** \[28\]. There are well-known differences in how experts and novices process diagrams \[22\].

It is good practice to involve “users” in the analysis and design of business processes; this also works in developing software systems (e.g. user-centered design) or in developing new products. Why not involving them in the design process of notations? Caire at al. \[27\] have done this for example in a research study regarding requirements engineering (RE) notations (i.e. \[i^\circ\]).

The key to designing visual notations that are understandable to naive users is a property called semantic transparency. This means that the meaning (semantics) of a symbol is clear (transparent) from its appearance alone; Semantically transparent symbols reduce cognitive load because they have built-in mnemonics \[29\] (see Figure 6).

However, semantic transparency is typically evaluated subjectively: experts (researchers, experts from software vendors) try to estimate the likelihood that novices will be able to infer the meaning of particular symbols. Even when notations are specifically designed for communicating with business stakeholders, members of the target audience are rarely involved. For example, BPMN 2.0 is a notation designed for communicating with business stakeholders, yet no business representatives were involved in the notation design process.
Semantic transparency is a continuum. [27]

Business process models are needed to facilitate a shared understanding in the organization; therefore the process creating and documenting the model includes employees unfamiliar with the chosen process design method. Typical workshops on process design employ design tools such as whiteboards, flip charts and post-its to capture knowledge about a current or future process. Informal sketches and diagrammatic drawings were found to be key to any design activity, as they serve as an externalization of one’s internal thoughts, and assist in idea creation and problem-solving.

There is a clear difference how novice and expert modelers conceptualize important domain elements as reported by Wang and Brooks [31], who found that novice modelers conceptualize in a fairly linear process in contrast to experts, who have better analysis and critical evaluation skills. Also based on unexperienced modelers, Recker et al. [32] developed a range of typical process design archetypes; they found out, that “moderate use of graphics and abstract shapes to illustrate a process is more intuitive and would aid the understanding on the concept of process modeling”.

III. Application

Our focus is on agent-based modeling and execution of business processes. The execution of agent-based business process models has already be discussed in [33] [13] [14]. The following chapter will discuss the modeling aspects of agent-based business process models based on the S-BPM methodology. Finally we will discuss ongoing work to find alternative and more user-friendly ways to develop process models.

A. Subject-oriented BPM

The S-BPM language [3], as supported in our execution platform [13] [14], is depicted in Figure 7 and Figure 8.

The Subject Interaction Diagram (SID) defines the Subjects (1) and the unidirectional Channels (4) between them. These channels establish the communication between the subjects and enables to send and receive messages at runtime. A Multi-subject allows to send a message to more than one agent (an agent is an instance of a subject); an External-subject allows to model a subject without knowing the internal behavior, for example another choreography which participants are not part of the own organization.

The Subject Behavior Diagram (SBD) defines the internal behavior of a subject; Send (1), Receive (2) and Action (3) are the fundamental activities for this diagram. The internal behavior of subjects has a minimum of one Start (4) and one End (5) activity. Any activity is marked with a flag to denote it as start or end activity. The normal control flow is defined as explicit Transition between activities. Timeout Transition are based on a relative time and model exceptional behavior to prevent dead lock situations or service level problems in case of no answer in a defined timeframe.

Contrary to BPMN 2.0, it can be easily seen that this notation has a 1:1 fit between concepts and semantic constructs (ontological analysis). It also corresponds with the notion of system (communicating agents) and models the state and transition space (ontological model). Nevertheless, at first glance it seems to be much simpler and therefore cognitively more effective; the advantage of easily bridging the gap (from model to execution) is evident, but there seem to be (hidden) mental hampering factors in the field of modeling, as experience shows. One could be, for example, that current software implementations do use two different views as elaborated above.
Other cognitive hampering factors have yet to be investigated.

B. Alternative Ways to Model

Recently Fleischmann \[34\] has presented a haptic way to model S-BPM processes. The main idea is to use building blocks to model the processes (Build Book), as can be seen in Figure 9.

The core concept now has to be transferred onto a technology platform; we propose to do the modeling process on a touch device\[.\] The technology platform should be as flexible as possible, as there are many unclear requirements yet.

1) Actual S-BPM Functionality: In our application, we utilize the screen of a tablet device and allow the user to drag and drop items from a library section to a stage section. Manipulation includes moving one or several items, selecting one or several items, connecting items, deleting one or several items, and increasing/decreasing item width as well as some convenience functionality if dealing with large process models. This includes zooming, moving and an automatically resizing of the stage.

Items are able to change their color or to get a background image (icon). Furthermore they can be labeled with text and are capable of holding properties, which are implemented as a list of key value pairs.

There are two ways of connecting items with each other. One possibility is the implicit method by concatenating blocks as described. Another possibility to create a connection between items is to use arrows. This is necessary when the flow is not sequential and needs to skip certain elements. How items are connected does not change their behavior, to illustrate this Figure 10 shows both ways of connecting items.

C. Block Modeler

Nevertheless, we think, that the use of blocks offers a very convenient way to model business processes, especially using a modeling notation, such as S-BPM as there are only a very limited number of symbols needed. The core visual ideas of our Block Models therefore are:

- all symbols are rectangles (blocks)
- the semantic and syntax is defined using different colors
- the blocks are laid out on a canvas
- the blocks are directly connected on one side; the flow direction is defined by convention (top-down, left-right or vice versa)
- to add additional needed directed connections, arrows can be used

The editor app is depicted in Figure 11. Two shades of gray indicate the two main areas of the editor. Dark gray represents the library with the available symbols. The lighter shade of gray which spans the biggest area in the application is what is internally named the stage. The stage is an UI element of type canvas, a canvas allows to have elements which can be arbitrary placed on it. The canvas can be zoomed in and elements can be moved around.

The intention of the modeling app is the need to have a flexible tool for further research and elaboration of the block modeling methodology. Anyway, we have to consider, that a fully functional modeling tool leads to many questions to be answered – especially if we want to find better ways to involve users in the modeling process.

The choice of platform allows to design a very interactive behavior with strong visual feedback mechanisms (in further

\[4\]http://www.metasonic.de/en/touch
development steps we also plan to implement the simulation of “physical” effects). For example, there is a “docking” mechanism, so overlapping blocks “jump” when released into the nearest valid position forming a connection between the blocks; alternatively it is also allowed to position blocks anywhere on the canvas (e.g. to have more than one model on it). It is also convenient to move models or parts of models around on the canvas.

As we could not find a ready to use library with classes to draw orthogonal directed arcs between blocks, we have to implement some basic algorithms for this, as depicted in Figure 12.

It is now possible to create a S-BPM process in analogy to the process shown in Figure 9. The subject interaction view is depicted in Figure 13 we implemented the concept of hierarchical modeling, so it is possible to drill down into the internal behavior of a subject; this is depicted in Figure 14. Messages between subjects can actually be added as key-value pairs. After that, the process can be persisted as XML-file and uploaded for execution.

There is one main problem: how to define the direction of flow or relations? We need a well defined and transparent syntax to reflect this – without using conventions (from left to right, for example). Finally, all models need to be persisted and it must be clear from the graphical representation which elements are connected, including a possible direction (typically visualized as directed arc).

2) Semantic- and Syntax Free Modling: An interesting point now is, that the concept of block modeling can be generalized towards a – as we call it – semantic and syntax free modeling method. The idea is as follows: people involved in any modeling problem could use the block modeling method to define a problem or domain specific semantic and syntax. Afterwards they could use this defined semantic and syntax to create a domain specific model. Using the concepts of ontology and the principles for designing cognitive effective visual notations this could improve the capabilities of people for conceptualization; the use of a touch screen further enables collaborative and interactive work (in small groups). In this view, S-BPM is only one possible language which can be used to create block models.

IV. DISCUSSION

Latest research proved that symbols designed by naïve subjects increase comprehensibility by a factor of almost
4 and reduce interpretation errors by more than 80% over notations designed in the traditional way. The average semantic transparency of novice-generated symbols was more than 5 times that of expert-generated symbols. Further on, semantic transparency significantly increases recognition accuracy and reduces interpretation errors. [27]

The most difficult attribute of business process models is the fact, that we try to visualize dynamic behavior using static visualizations; another one is to use a plethora of different symbols. The concept of block models opens some interesting questions with the intention to better understand the way how people, novice to modeling in general and especially inexperienced to model semantic precise business process models, can improve their related capabilities.

To this date there are some serious open questions, but in our opinion the block modeling methodology offers a promising research direction to understand cognitive hampering factors to understand process models.

The idea of a syntax and semantic free editor further on leads to the following research questions we will tackle in the future:

• Is it possible and useful to enhance the syntax according the full set of visual variables [19]?
• How to store the semantic and syntax in the modeler, i.e. definition of the modeling language (conformance check during modeling etc.)?
• Is it possible to find a cognitive easy method to visualize flow and relations (beside the typical use of directed arcs)?

These research has to go along with studies of modeling in the field; i.e. to support inexperienced people to learn how to conceptualize in general and how to create business process models. This is supported by the use of a tool, which can be adapted as learning increases.

Another interesting question could be, if it is possible to model instances instead of process models, which include all possible instances; a software could merge all possible instances into a general model, a blue print of a business process. This is only possible using notations which are mathematically precise, as the S-BPM method.

REFERENCES

[1] “Gartner BPM Hype Cycle,” Gartner, Tech. Rep., 2012.
[2] E. Aitenbichler, S. Borgett, and M. Mülhäuser, “Distributed Execution of S-BPM Business Processes,” in S-BPM ONE 2010. Berlin, Heidelberg: Springer, 2011, pp. 19–35.
[3] A. Fleischmann, W. Schmidt, C. Stary, S. Obermeier, and E. Börger, Subject-Oriented Business Process Management. Springer, 2012.
[4] Enabling Customer Experience and Front-Office Transformation through Business Process Engineering. IEEE Conference on Business Informatics. Vienna: Key Note, 2013.
[5] E. Börger, “Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL,” Software & Systems Modeling, 2011.
[6] J. Sinur, J. Odell, and P. Fingar, Business Process Management: The Next Wave, ser. Harnessing Complexity with Intelligent Agents. Meghan-Kiffer Press, 2013.
[7] A. Fleischmann, “What is S-BPM?” in S-BPM ONE - Setting the Stage for Subject-Oriented Business Process Management. Springer, 2010, pp. 85–106.
[8] ——, Distributed Systems: Software design and implementation. Springer, 1994.
[9] ——, PASS - A Technique for Specifying Communication Protocols. North-Holland Publishing Co., May 1987.
[10] R. Milner, Communication and Concurrency. Prentice-Hall, 1989.
[11] M. Wooldridge, An Introduction to MultiAgent Systems. John Wiley & Sons, Jun. 2009.
[12] M. Ben-Ari, Principles of Concurrent and Distributed Programming. Pearson Education, 2006.
[13] S. Raß, J. Kotremba, and R. Singer, “The S-BPM Architecture: A Framework for Multi-agent Systems,” 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 3, pp. 78–82, 2013.
[14] R. Singer, J. Kotremba, and S. Raß, Modeling and Execution of Multi-enterprise Business Processes,” in Conference on Business Informatics.
[15] Y. Wand and R. Weber, “An ontological model of an information system,” IEEE Transactions on Software Engineering, vol. 16, no. 11, pp. 1282–1292, 1990.
[16] E. D. Falkenberg, “A Framework Of Information System Concepts,” Tech. Rep., 1998.
[17] J. L. G. Dietz, Enterprise Ontology: Theory and Methodology. Springer, 2006.
[18] M. Bunge, Ontology: The Furniture of the World, ser. Treatise on Basic Philosophy. Reidel, 1977, vol. 3.
[19] J. Bertin, Semiology of graphics, ser. diagrams, networks, maps. esri press, 1983.
[20] J. H. Larkin and H. A. Simon, “Why a Diagram is (Sometimes) Worth Ten Thousand Words,” Cognitive science, vol. 11, no. 1, pp. 65–100, 1987.
[21] D. Moody, “The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering,” IEEE Transactions on Software Engineering, vol. 35, no. 6, pp. 756–779, 2009.
[22] P. Cheng, R. K. Lowe, and M. Scaife, “Cognitive science approaches to understanding diagrammatic representations,” Artificial Intelligence Review, 2001.
[23] A. Gehlert and W. Esswein, “Toward a formal research framework for ontological analyses,” Advanced Engineering Informatics, vol. 21, no. 2, pp. 119–131, Apr. 2007.
[24] S. Gregor, “The nature of theory in information systems,” MIS quarterly, 2006.
[25] C. E. Shannon and W. Weaver, The Mathematical Theory of Communication. Univ. of Illinois Press, 1963.
[26] R. Singer and E. Zinser, “Business Process Management - Do We Need a New Research Agenda?” in Subject-Oriented Business Process Management. Springer, 2011, pp. 220–226.
[27] P. Caire, N. Genop, and P. Heymans, “Visual notation design 2.0: towards user comprehensible requirements engineering notations,” Requirements . . . , 2013.
[28] C. Heath and D. Heath, Made to Stick, ser. Why Some Ideas Take Hold and Others Come Unstuck. Arrow Books Limited, 2007.
[29] M. Petr, “Why looking isn’t always seeing: readership skills and graphical programming,” Communications of the ACM, vol. 38, no. 6, pp. 33–44, Jun. 1995.
[30] J. Recker, “Opportunities and constraints: the current struggle with BPMN,” Business Process Management Journal, vol. 16, no. 1, pp. 181–201, 2010.
[31] W. Wang and R. J. Brooks, “Empirical investigations of conceptual modeling and the modeling process,” in 2007 Winter Simulation Conference. IEEE, pp. 762–770.
[32] J. Recker, N. Safrovin, and M. Rosemann, “How Novices Model Business Processes,” vol. 6336, pp. 29–44, 2010.
[33] J. Kotremba, S. Rall, and R. Singer, “Distributed Business Processes - A Framework for Modeling and Execution,” Sep. 2013.
[34] C. Fleischmann, “Subject-oriented Process Survey,” Ph.D. dissertation, Vienna University of Technology, Nov. 2013.