



HAL
open science

Supporting Modelers with Model Qualities and Patterns

Fabrice Boissier

► **To cite this version:**

Fabrice Boissier. Supporting Modelers with Model Qualities and Patterns. 11th IEEE International Conference on Research Challenges in Information Science, May 2017, Brighton, United Kingdom. hal-01523922

HAL Id: hal-01523922

<https://hal.science/hal-01523922>

Submitted on 17 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Supporting Modelers with Model Qualities and Patterns

Fabrice Boissier

Centre de Recherche en Informatique, Université Paris 1 - Panthéon Sorbonne
90, Rue de Tolbiac, 75013, Paris, France
Fabrice.Boissier@univ-paris1.fr

Abstract—Modelers face multiple challenges in their work. In this paper, we focus on two of them. First, multiple modeling methods and tools are currently available. Modelers are sometimes limited by their tools or paradigms. Second, when multiple models are proposed for the same case, a decision maker needs criteria to decide which model to choose for his/her objective.

This Ph.D thesis aims to explore designer questions for creating the best model.

For the first steps, a case study on design patterns applied to modeling process will be made, and a BPMN concepts related comparison function will be proposed.

Keywords—Models, Modeling Process, Quality of Model, Models Comparison, Model Semantic, Process Mining, Model Patterns.

I. INTRODUCTION

The modeling activity is used in multiple fields of study for transposing a complex problem into a simpler representation using abstract components. The granularity of the model depends on the choice of the modeler to take into account few or more details from each component. With less details, some components become obvious, and problems can be easily solved. In contrary, too many details increase the complexity of a problem, its solving time, and also the modeling time, but, it becomes more realistic.

While companies use a lot of models within their ERP [1], merging two companies become a difficult task when too many processes are currently used. Challenges of models redundancy also appear in big companies that reorganize. Choosing the best model to keep between multiple options requires a human decision.

Current process model comparison methods are using multiple metrics like structural similarity, behavioral similarity, and syntactic similarity. Structurally, these methods compare the nodes, edges, and words inside labels. In behavioral metrics, the nodes order is important. The syntactic metric is often re-used inside the two previous metrics to obtain more details about context, and it is based on word equivalence, synonyms, and/or Levenshtein distance. These metrics try to find the behavior from the model structure and the words composing its annotations or labels [2] [3] [4].

This paper aims to define a Ph.D thesis for exploring model designers questions through multiple application domains (paradigm, concepts, architecture, granularity level).

Objectives concern particularly two parts of the modeling activity: the simplification of modelers work (model building), and the simplification of decision making when multiple models representing the same system or process are proposed (model usage). Modelers' work can be eased with techniques used as best practices in computer science, like the *design patterns* [5]. When multiple models representing the same system or process are available, it is somewhat difficult to decide which one should be used. For this reason, the application domain of the model components are analyzed to determine the set of concepts used, and help a decision maker to choose the model containing the required concepts. As a first step, a case study on a young modeling framework is conducted to identify modeling difficulties and propose some design patterns to help future models building. Next, a plan for comparing business processes with their embedded concepts and find the most suitable depending on the situation is presented.

The paper is organized as follows. In the first part, the context was established. In the second, the related work around models qualities and business models comparison will be presented. In the third part, questions and objectives will be explained. In the fourth part, a roadmap for the thesis will be shown. In the fifth part, an experiment with the PyCATSHOO modeling framework [6] is presented, and a future work on the comparison of BPMN labels concepts is described.

II. RELATED WORK

Multiple criteria are proposed in the literature to explain qualities of models. In the conceptual modeling community, [7] uses two modeling frameworks (LSS for Lindland, Sindre, and Sølvsberg [8], and BWV for Bunge-Wand-Weber [9]) to build the CMQF (Conceptual Modeling Quality Framework) based on four layers (physical, knowledge, learning, development).

First, the LSS framework concentrates on the conceptual modeling product, and defines qualities dimensions (syntactic quality, semantic quality, perceived semantic quality, pragmatic quality, social quality, physical quality, and empirical quality). These dimensions measure if the representation is valid, exhaustive, well-built depending on the modelers knowledge, and clearly understood by the target audience.

Second, the BWW framework concentrates on the conceptual modeling process, and defines steps (application domain, information system, modeling grammar, ontological construct, user's view of domain, user's view as inferred from information system). The steps guide the modeler to build an information system by first getting the domain of application, then classifying things of the real world and conceptualize them, and finally by getting user's view of the domain with and without the information system created.

These frameworks are reorganized in two dimensions to obtain the CMQF and its four layers. The physical layer contains every element of the framework that can be physically see by a person. This layer contains seven qualities (model-domain appropriateness, ontological quality, syntactic quality, semantic quality, language-domain appropriateness, intensional quality, empirical quality). The knowledge layer concerns the representation of things in the mind of people and their qualities (perceived model-domain appropriateness, perceived ontological quality, perceived syntactic quality, perceived semantic quality, perceived language-domain appropriateness, perceived intensional quality, perceived empirical quality). The learning layer is measuring how well things are perceived when someone tries to acknowledge them with four qualities (view quality, pedagogical quality, linguistic quality, pragmatic quality). The development layer measures if the knowledge has been correctly used to build the physical representation. It is based on six qualities mainly linked with the application domain and language to express knowledge (applied domain - model appropriateness, applied domain - language appropriateness, applied domain knowledge quality, applied model - language appropriateness, applied model knowledge quality, applied language knowledge quality). The CMQF framework allows to measure how well the representation is, how well things are represented in the mind of people, how well information can be learned, and how well knowledge has been expressed in the representation.

Literature on business models comparison shows multiple techniques.

The simple method of comparison for graph-based models is instinctively the comparison of the structure (nodes and links). Various papers explored better ways of comparing these types of models. For example, in [10], the comparison is based on the traces of the model, and particularly the *typical behavior* of the process. It is done by reading the traces of the process in the event log. The authors also stated that the bisimilarity method is too strict, because it answers if two processes are exactly the same (true) or not (false), and no *distance* are available.

In [11], four algorithms making structural and labels comparisons are used. The approach is based on graph-edit distance (requiring graph traversal) following rules: similarity greedy traversal, maximal pruning, heuristic pruning, and A*. Labels are also compared using semantic and syntactic distance. The main semantic measure used is based on words comparison and synonyms. Two words are equal (1), or

synonyms (0.75), or different. The syntactic difference is based on the number of atomic operations required to transform one word into another one (insertion of one character, deletion, change).

In a similar approach, the papers [4] [3] are using the word-based semantic and syntactic distances of the labels with the structure (like [11]), and they introduce the *causal footprint* [12] in the method. The *causal footprint* is a matrix containing information about a model (or traces) concerning its nodes, links, and their order. Instead of numbers, the matrix contains symbols expressing if one node appears directly before (\leftarrow), directly after (\rightarrow), or in parallel (\parallel) with another node, or if it's not a direct neighbor ($\#$). The causal footprint gives a similarity value between two models (or between a model and traces) by checking the differences within their matrices. The papers use the causal footprint as a behavior similarity metric, and the labels name as a syntactic similarity metric. [4] compares the structure with a graph-edit distance and the labels name, but [3] uses a context similarity metric based on a specificity of EPC (Event-driven Process Chain, a modeling method) where each function (an activity node) is surrounded before and after by events (context nodes). This specificity induced by the paradigm ease the work of gathering context (the predecessor and the successor nodes give the context), but the comparison method presented only works for EPC.

III. RESEARCH QUESTIONS AND OBJECTIVES

The main objective is to help organizations in building and using the best models depending on the situation they are facing. Qualities are required to better understand what makes a model more attractive than another one. Typical expected qualities concern complexity level of the modeling process, ease of understanding, maintainability. Exploration around paradigm, concepts, architecture, and granularity level is required. This leads to broad research questions about what makes good models:

- *Which criteria or metrics are important when building a model?*
- *Which criteria or metrics should be used to choose the best model between multiple?*

To focus more on the decision process when multiple models are available, model comparison is explored. Literature is mainly concentrated on business models (with graph-based models) for this topic and uses multiple metrics based on structure, behavior, and labels. Current criteria are analyzing models by comparing structure (nodes order) and semantic (labels comparison). Semantic is currently analyzed using word-based comparison between the labels of each model, which makes a local criteria of comparison. To help a decision maker to choose the best fitting model depending on its application domain and users, a more global criteria could be proposed using the concepts contained within labels. If specifications are given, it may also be possible to tell if a model is exhaustive or

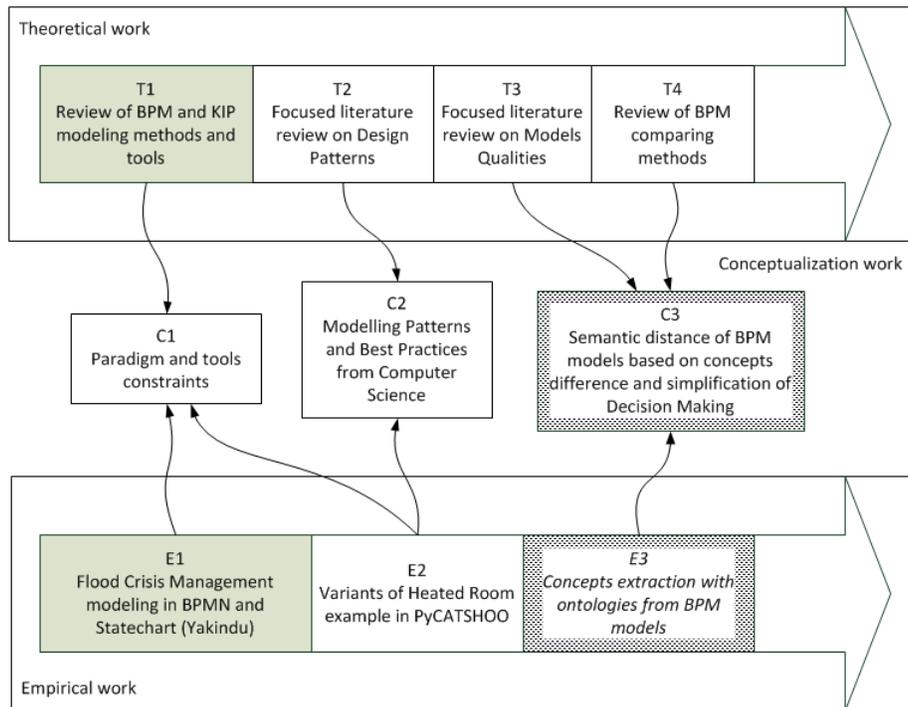


Figure 1. The roadmap describing our past, ongoing and prospective research for the first steps of the thesis.

not concerning its domain of application. This leads to more specific questions around label semantic:

- *Do the BPM model labels embed the application domain concepts of the whole model?*
- *Is it possible to measure the completeness of a BPM model using its labels embedded concepts?*
- *Is it possible for a decision maker to use the difference of concept set as a criteria of choice?*

Modeling with much details might be mandatory in some cases, then, it becomes a long and somewhat difficult task (depending on the paradigm, the meta-model chosen and its constraints). The use of modeling patterns, based on previous cases, should reduce the time and difficulty of this activity. A possible threat appears when a pattern replaces better representations: a less obvious pattern is used to illustrate something, losing some ease to understand, but, time is gained while modeling, considering the proposed patterns may be based on previous cases with some semantics attached to it. Practical questions about modeling patterns in the applicability and usability of modeling tools come next:

- *Is it possible to propose some predefined patterns with the simultaneous objectives of reducing the effort of modeling, gaining time, and upgrading the understanding?*
- *How the patterns should be collected and proposed to the modeler?*

Another possible way of using modeling patterns is by deriving the design patterns [5] where recurrent problems are solved by standard solutions. These modeling patterns would

also carry some knowledge about problems and solutions. As they embed informations, it becomes possible to find back the design goal. However, some questions about this type of patterns are raised:

- *Do modeling pattern carry semantics about problem and/or solution?*
- *Do modeling patterns embed business informations? Or is it modeling technical informations?*
- *How this semantic could be used for a modeler to ease his work?*

IV. ROADMAP

A roadmap for the thesis first steps is presented in Figure 1. The steps are classified between theoretical work (aggregation of theoretical work from literature), empirical work (experimentation to test our hypotheses), and conceptualization (deduction of new knowledge from experimentation and literature). Grey background rectangles stand for achieved task, white background rectangles stand for current work, and bold bordered rectangles concern future work.

First, in step T1, a review of the current tools for modeling BPM was required to better understand the strengths and weaknesses of the current tools. In parallel to this step, the task E1 was a modeling experiment of a flood crisis management process using official documentation ¹. Multiple models were made in MAP [13], BPMN [14], and Statecharts [15] [16] to

¹Russian COSOC, French PCS (Plan Communal de Sauvegarde), French DDRM - Risque Inondation en Île de France, and French plan ORSEC Île de France

check modeling possibilities and limits. The step C1 is the result of these two first steps: the regular BPMN is unable to support unpredictable events or decisions, and transitions cannot happen *because* of an external event. The experiment results were published in previous works [17] [18] [19].

To help modelers in building the best models, the step T2 studies the IT development best practices where Design Patterns [5] are used as basis to recurrent problems. During step E2, an experiment is made in collaboration with EDF R&D² and one of their tool (PyCATSHOO [6], a stochastic automata simulator) to propose various models of a single example case (heated room) and extract some patterns for their modelers. The results of these combined steps is a set of useful patterns for modelers in C2. As the tool is following a precise paradigm and rules, some constraints for the modelers are also extracted in C1.

Step T3 collects current knowledge regarding qualities of models, and step T4 analyzes the current techniques of BPMN comparison. A comparison function will be produced in E3 to compare BPMN models based on the concepts embedded within their labels. An additional criteria of semantic distance, based on model concepts' completeness, is expected to be produced in C3. This will help decision maker by giving them an additional information about each model, and which one is more appropriate to which users.

V. CASE STUDY AND MODEL COMPARISON

A. PyCATSHOO Case Study

A case study is currently under work on the modelers practices with a modeling framework code-driven.

PyCATSHOO [6] is a tool developed by EDF in its Research and Development branch, for better modeling risks and failures in their power plants and dams. The main asset of PyCATSHOO is its capacity to manage simultaneously discrete probabilities and continuous state inside automata. A transition between two states happens following precise events, or, after a certain amount of time depending on the probability given to an event. The models are built with classes which communicates using multiple message boxes (see Figure 2). Each class contains multiple components (see Figure 3):

- stochastic automata using states and transitions (some deterministic transitions happen when a limit is reached, some stochastic transitions happen depending on a probability),
- optionally, a sub-automaton (called PDMP controller, for Piecewise Deterministic Markov Processes) is used to support the continuous state transitions using equations (typically to model a flow, and decide when the limit is reached for activating a transition in the main automaton attached to it),
- internal and external variables managed by states and transitions, and linked to message boxes,

²EDF - Électricité de France, french national energy producer and distributor

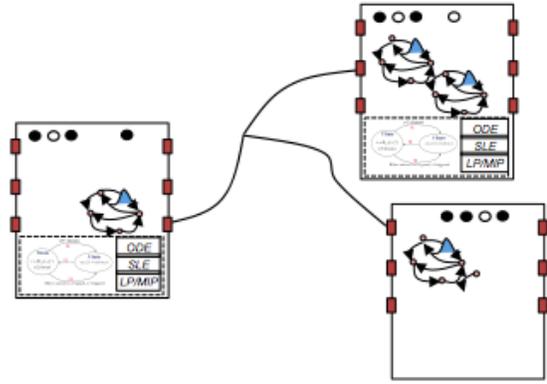


Figure 2. Multiple PyCATSHOO classes linked together with message boxes.

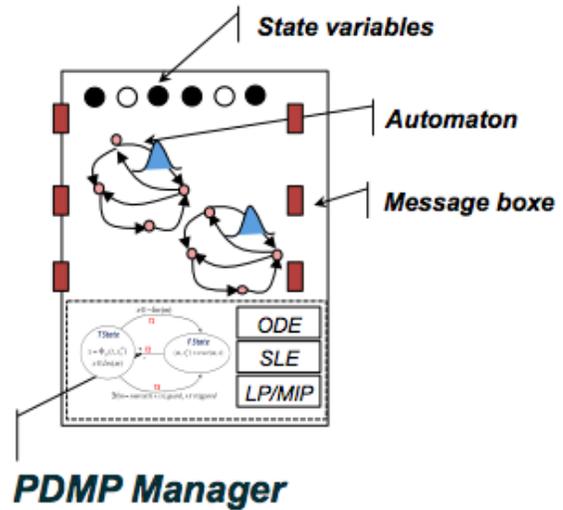


Figure 3. The internal component of a PyCATSHOO class.

- message boxes for sending and receiving messages (variables states) to the other classes.

A specificity of the generated models is that each class generally owns a *failure* automaton (the object is broken and blocked in a certain state) associated to a *time to repair* probability, and a *regular work* automaton (the regular states where the object can be).

The collaboration with EDF on PyCATSHOO aims to help their modelers to create better models or use a better modeling method. Initially, EDF was using fault tree analysis for predicting the most probable cause of a failure and calculate risks. This approach shows some limits when the combination of some events was unexpected. Researchers at EDF changed the way the models were built and run thanks to PyCATSHOO. Instead of beginning from an hypothesis "keep at least one critical path of cooling circuits" and calculate probabilities on each part, they start from a business goal "do not achieve nuclear meltdown" which translates into "do not reach a precise temperature" resulting from water flow and heat exchange of

cooling circuits (typically, a continuous state). PyCATSHOO is still a young tool with few best practices for now, but it is promising for the risk assessment and modeling domains.

A first experiment is conducted where one of the example models given with PyCATSHOO (an Heated Room model) is re-used to produce models under new points of view and produce model patterns (based on the design patterns [5] idea).

The Heated Room example is currently built using one "Heater" class, and one "Room" class where the heat is supported by a PDMP. The two classes are linked with two message boxes : sensor and room.

The initial example is transformed to support multiple heaters working in a backup mode with various priorities order. Two implementations are proposed where one is built with the initial paradigm, and one with a design pattern which completely removes the message boxes between classes. The pattern is reducing the number of links to write as a modeler, but it introduces a concept that is coming from the computer science domain.

As PyCATSHOO modelers are experts in risk assessment, but not in software engineering, the introduction of design pattern for models might be complex for modelers, but it is expected to be efficient for huge cases where the number of classes would make an exponential number of links. The complexity of the modeling pattern to be introduced will be measured with various code review tools, and compared with the complexity of the regular implementation without design pattern. Some constraints are also already collected in the two cases: typically, some code needs to be duplicated for expressing a transition and a stop condition in the PDMP evolution.

These remarks open a promising view on the introduction of computer science best practices for models based on code to help modelers in their work.

B. BPMN Model Comparison

BPMN model comparison is currently using label comparison to check if words are similar, synonyms, or different [11] [4] [3]. This method allows a local comparison between ordered activities. The result is a global mark giving the distance between two models. A decision maker does not have enough criteria to decide which model to keep simply with a distance degree. This experiment aims to produce an artifact that will create a set of concepts used inside a BPMN model. The concepts will help the decision maker to choose a more eligible model depending on the situation or the audience. For example, a model giving details about *engineering pricing per days* and *task realization* will be more adapted to a project manager than a human resource employee who will prefer a model which uses *days of absence* and *employees cost*. This artifact would extract global knowledge about a model: what are the inside embedded concepts.

My proposed artifact will be a function of concepts extraction. The *concepts* are ontological categories to which the

labels words (or expression) are members. Typically, *task realization* (label) is owned by the *project management* (category) application domain. The concepts and their semantic links with the application domain were explored in the conceptual modeling frameworks LSS [8], BWW [9], and CMQF [7]. The CMQF and BWW describe multiple qualities in which the *applied domain* of the representation should match with the work domain of stakeholders and things to be modeled. Similarly, the LSS framework describes the *Semantic Quality* where the representation should correspond to the domain by checking the validity and completeness of assertions. These qualities are the criteria which will help a decision maker to choose the best fitting model within multiple existing ones.

To design this function, a design science methodology [20] [21] is chosen. The general methodology proposed would be to use existing or adapted BPM models with the exact same concepts for some of them, and slightly different concepts for others, build the function which will extract concepts and compare them, and finally evaluate the success and failure rates for concepts matching of the function depending on the models. The evaluation will be based on the concepts extracted by a human (a modeler or an expert of the domain) and compared with what the function will extract.

To build the function, it will be required to clearly understand how the previous papers are making comparison between models, and more precisely between labels. The extraction of concepts will be based on ontologies to match realistic categories. This idea can be extended in the case of organizations applying the enterprise architecture method 4EM [22]. The 4EM framework relies on a *concept model* that is used to create every business process of an organization.

VI. CONCLUSION

This paper gave a first roadmap for a Ph.D thesis. It describes the exploration of two modeling practices challenges:

- 1) how to help modelers with design patterns,
- 2) how to help decision maker to choose the more adapted model between multiple options considering its audience and application domain, and using a comparison function.

REFERENCES

- [1] M. Rosemann, "Potential pitfalls of process modeling: part a," *Business Process Management Journal*, vol. 12, no. 2, pp. 249–254, 2006.
- [2] M. Kunze, M. Weidlich, and M. Weske, "Behavioral similarity—a proper metric," in *International Conference on Business Process Management*. Springer, 2011, pp. 166–181.
- [3] B. Van Dongen, R. Dijkman, and J. Mendling, "Measuring similarity between business process models," in *Seminal Contributions to Information Systems Engineering*. Springer, 2013, pp. 405–419.
- [4] R. Dijkman, M. Dumas, B. Van Dongen, R. Käärrik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Information Systems*, vol. 36, no. 2, pp. 498–516, 2011.
- [5] E. Gamma, *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.
- [6] H. Chraïbi, "Dynamic reliability modeling and assessment with pycatshoo: Application to a test case," in *Proceedings of PSAM*, 2013.
- [7] H. J. Nelson, G. Poels, M. Genero, and M. Piattini, "A conceptual modeling quality framework," *Software Quality Journal*, vol. 20, no. 1, pp. 201–228, 2012.

- [8] O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding quality in conceptual modeling," *IEEE software*, vol. 11, no. 2, pp. 42–49, 1994.
- [9] 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.
- [10] W. M. van der Aalst, A. A. De Medeiros, and A. Weijters, "Process equivalence: Comparing two process models based on observed behavior," in *International Conference on Business Process Management*. Springer, 2006, pp. 129–144.
- [11] R. Dijkman, M. Dumas, and L. García-Bañuelos, "Graph matching algorithms for business process model similarity search," in *International Conference on Business Process Management*. Springer, 2009, pp. 48–63.
- [12] B. Van Dongen, J. Mendling, and W. Van Der Aalst, "Structural patterns for soundness of business process models," in *Enterprise Distributed Object Computing Conference, 2006. EDOC'06. 10th IEEE International*. IEEE, 2006, pp. 116–128.
- [13] C. Rolland, N. Prakash, and A. Benjamen, "A multi-model view of process modelling," *Requirements Engineering*, vol. 4, no. 4, pp. 169–187, 1999.
- [14] OMG, *Business Process Model and Notation (BPMN), Version 2.0*, Object Management Group Std., Rev. 2.0, January 2011. [Online]. Available: <http://www.omg.org/spec/BPMN/2.0>
- [15] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of computer programming*, vol. 8, no. 3, pp. 231–274, 1987.
- [16] —, "On visual formalisms," *Communications of the ACM*, vol. 31, no. 5, pp. 514–530, 1988.
- [17] E. Kushnareva, I. Rychkova, R. Deneckère, and B. Le Grand, "Modeling crisis management process from goals to scenarios," in *International Conference on Business Process Management*. Springer, 2015, pp. 55–64.
- [18] E. Kushnareva, I. Rychkova, and B. Le Grand, "Modeling business processes for automated crisis management support: lessons learned," in *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*. IEEE, 2015, pp. 388–399.
- [19] —, "Modeling and animation of crisis management process with statecharts," in *International Conference on Business Informatics Research*. Springer, 2015, pp. 145–160.
- [20] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [21] R. H. Von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
- [22] K. Sandkuhl, J. Stirna, A. Persson, and M. Wißotzki, "Enterprise modeling," *Tackling Business Challenges with the 4EM Method*. Springer, p. 309, 2014.