Situational Business Model Developer: A Tool-support for Situation-specific Business Model Development

Sebastian Gottschalk
Universität Paderborn, Germany, sebastian.gottschalk@uni-paderborn.de

Enes Yigitbas
Universität Paderborn, Germany, enes.yigitbas@uni-paderborn.de

Alexander Nowosad
Universität Paderborn, Germany, anowosad@mail.uni-paderborn.de

Gregor Engels
Universität Paderborn, Germany, engels@upb.de

Situational Business Model Developer: A Tool-support for Situation-specific Business Model Development

Sebastian Gottschalk, Enes Yigitbas, Alexander Nowosad, and Gregor Engels

Paderborn University, Software Innovation Lab, Paderborn, Germany
{sebastian.gottschalk,enes.yigitbas.gregor.engels}@uni-paderborn.de
anowosad@mail.uni-paderborn.de

Abstract. The development of business models is a challenging task that can be supported with software tools. Here, existing approaches and tools do not focus on the company's situation in which the development takes place (e.g., financial resources, product type). To tackle this challenge, we used design science research to develop a situation-specific business model development approach that contains three stages: First, existing knowledge in terms of tasks to do (e.g., analyze competitive advantage), and decisions to be made (e.g., social media marketing) are stored in repositories. Second, the knowledge is used to compose a development method based on the company's situation. Third, the development method is enacted to develop a business model. This demonstration paper presents a tool-support called Situational Business Model Developer that supports all stages of our approach. We release the tool under open-source and evaluate it with a case study on developing business models for mobile apps.

Keywords: Business Model Development, Situational Method Engineering, Tool Support

1 Introduction

The development of business models, defined by Osterwalder et al. as "the rationale of how the organization creates, delivers, and captures value" [17], is an essential task for a company to remain competitive and sustainably successful. Studies of startups [6], as well as established companies [10], have shown the difficulty of developing effective business models. Here, the development can be guided based on the knowledge of domain experts. This knowledge can support the development method with existing processes [15] or repositories [2] of tasks to do (e.g., analyze competitive advantage) or the modeling with taxonomies [13] and patterns [8] of decisions to be made (e.g., social media marketing). However, both the development method for business model (i.e., later method) and the business model itself (i.e., later canvas model) need to fit the situation of the company (e.g., financial resources, addressed market size) [7] and the application domain of the product / service (e.g., mobile app, social network) [3]. In turn, this is not the focus of existing software tools [20, 4], which are primarily based on the business model itself and not the development method behind it (e.g., [1,5,19]).

17th International Conference on Wirtschaftsinformatik,
February 2022, Nürnberg, Germany

Therefore, we are conducting a design science research (DSR) study [15] to create a situation-specific business model development approach. By using DSR, we aim to provide a knowledge contribution [11] of nascent design theory of principles for such approaches and a corresponding architecture of the tool. By considering the knowledge contribution framework [11], we are doing this based on an exaptation, which means that we transfer an existing solution to a new problem domain. We are doing that by transferring the solution of situational method engineering [14] from software development to the problem of business model development. The approach [12] consists of the four roles of the *Domain Expert*, the *Method Engineer*, the *Business Developer*, and other *Stakeholders* together with the three stages of *(1) Knowledge Provision of Methods and Models*, *(2) Composition of Development Methods*, and *(3) Enactment of Development Methods*. At the *(1) Knowledge Provision of Methods and Models*, the *Method Engineer* formalizes the knowledge of the *Domain Experts* about the development method and the business model itself to allow its usage within the tool. Next, the *(2) Composition of Development Method*, the *Method Engineer* takes the context of the company (situation of the company and domain of the product/service) from the *Business Developer* and composes a situation-specific development method and chooses domain-specific models for some method steps. Finally, in *(3) Enactment of Development Methods*, the *Business Developer* enacts the development method and performs the proposed steps. At this step, the *Business Developer* can collaborate with the other *Stakeholders* during each step. In contrast to other business model development approaches, we point out the importance of the *Method Engineer* who is responsible for the knowledge structuring and development method composition. To support the approach, we are creating the *Situational Business Model Developer*, a software tool that can divide between the different roles and consists of all three stages.

The rest of the paper is structured as follows: Section 2 provides a component diagram of our tool architecture. Section 3 describes the implementation techniques and shows the three stages of knowledge provision, method composition, and method enactment. Section 4 discusses our evaluation and shows future work.

2 Solution Architecture

The software architecture (see Figure 1) can be divided into the *Database*, the *Method Modeler*, the *Canvas Modeler*, and the *Development Method Engine*. The *Database* is used to store the knowledge of *Methods* and *Models* together with the composed and enacted *Development Methods*. The *Method Engineer* uses the *Method Editor* to store the *Method Knowledge* from the *Domain Experts* in the *Method Repository* and the *Canvas Editor* to store the *Model Knowledge* of the *Domain Experts* in the *Canvas Repository*. After that, the *Method Engineer* captures the *Context* to construct the *Development Method* by using the *Method Composer* based on the situation of the company and the *Model Composer* based on the application domain of the product / service. The *Business Developer* enacts the *Development Method* and uses his own *Information* on the *Method Enactor* and the *Model Enactor*. Moreover, that *Information* can come from other *Stakeholders*. As a result, the *Business Model* is created.

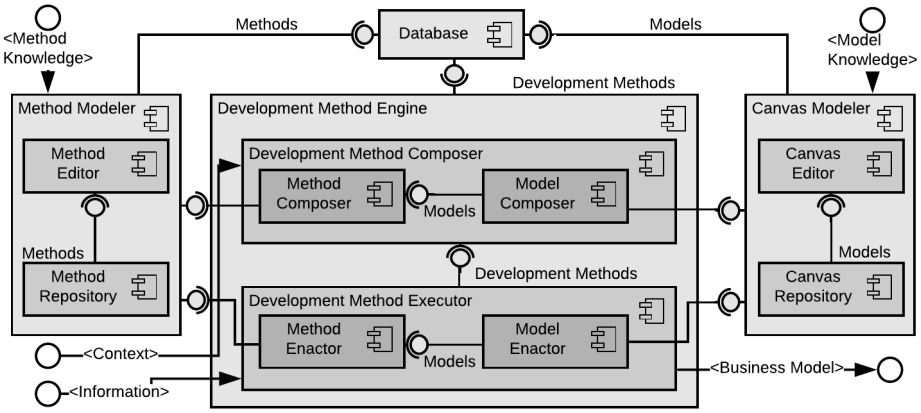

Figure 1. Overall Architecture of the *Situational Business Model Developer*

3 Solution Implementation

For our situation-specific business model development approach, we provide a tool-support called *Situational Business Model Developer*. The source code of the tool is released as open-source[1] and can be accessed in the web browser[2]. Here, we based the tool on the Angular framework[3] to allow an easy adaptation in the future. Moreover, we based our database on PouchDB[4] to run the tool standalone in the web browser. Last, we use BPMN.js[5] for the representation of the development methods. In the following, we focus on the method part as contrast to existing software tools. However, the canvas model part can be accessed through the tool and our research paper [12].

3.1 Stage 1: Knowledge Provision of Methods and Models

In the beginning, the *Method Engineer* needs to formalize the knowledge of the *Domain Experts* in a method repository and a canvas repository. Inside the method repository, there are method elements, method building blocks, and method patterns. Method elements are atomic parts of a development step divided into the tasks to do (e.g., conduct customer interviews), the situational factors (e.g., financial resources), types (e.g., discover), involved stakeholders (e.g., customer), used tools (e.g., prototyping tool) and responsible artifacts (e.g., prototype). Those elements are combined to method building blocks in the form of development steps (e.g., customer interviews with type discover and stakeholder customer) and arranged through method patterns. A method pattern is a BMPN model (see Figure 2 A), where a placeholder type for every activity

[1] Source Code: https://github.com/SebastianGTTS/situational-business-model-developer
[2] Online Version: http://sebastiangtts.github.io/situational-business-model-developer/
[3] Website of Angular Framework: https://angular.io/
[4] Website of PouchDB: https://pouchdb.com/
[5] Website of BPMN.js: https://bpmn.io/toolkit/bpmn-js/

is specified (see B), types can be defined (see C), and situational factors can be set (see D). Inside the canvas repository, there are canvas elements, canvas building blocks, and canvas models. Canvas elements are atomic parts (e.g., subscription) that are typically placed as sticky notes on a canvas. Canvas building blocks arrange those elements into trees for refinement (e.g., subscription to monthly subscription) with crosstree dependencies (e.g., in-app advertisement hurts privacy) and application domains (e.g., mobile apps). Canvas Models visualize part of those building blocks (e.g., Value Proposition Canvas, Business Model Canvas) on well-known canvas structures.

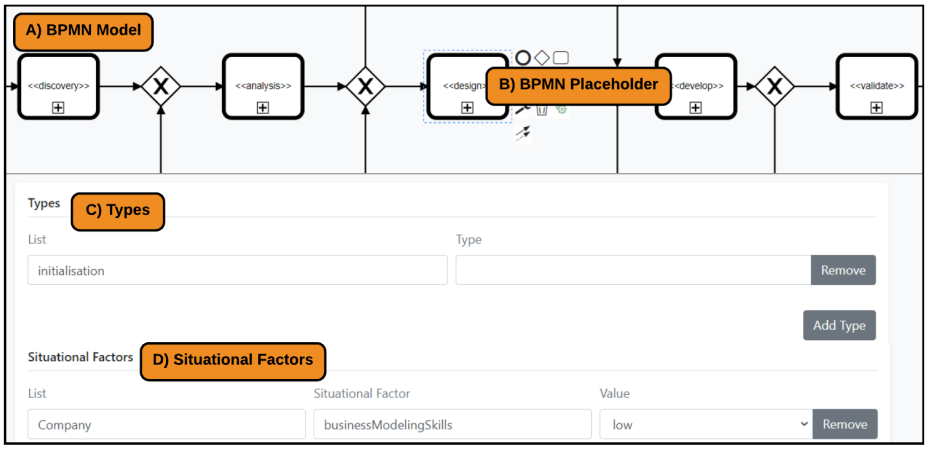

Figure 2. Creation of Method Patterns

3.2 Stage 2: Composition of Development Method

After that, the *Method Engineer* needs to compose the development method out of the provided knowledge. Here, the engineer interviews the *Business Developer* to get the purpose of the development, the situational factors of the company, and the application domain of the product/service. The engineer inserts this context in the tool and receives a development method as empty BPMN model. Inside the development method, he can insert the first method pattern. After that, the engineer can fill the corresponding placeholders with other method patterns or method building blocks of the placeholder type. Both the patterns and the building blocks are recommended based on the predefined situational factors. Moreover, the conformance of the whole development method is checked (see Figure 3 F) to see if all method parts fit to the context and all placeholders are filled out. Finally, after the method is constructed, the engineer can add a canvas model with predefined knowledge to each development step as artifact. For that, the engineer clicks on the step, chooses the specific canvas model, and receives canvas building blocks as recommendations based on the application domain (see G).

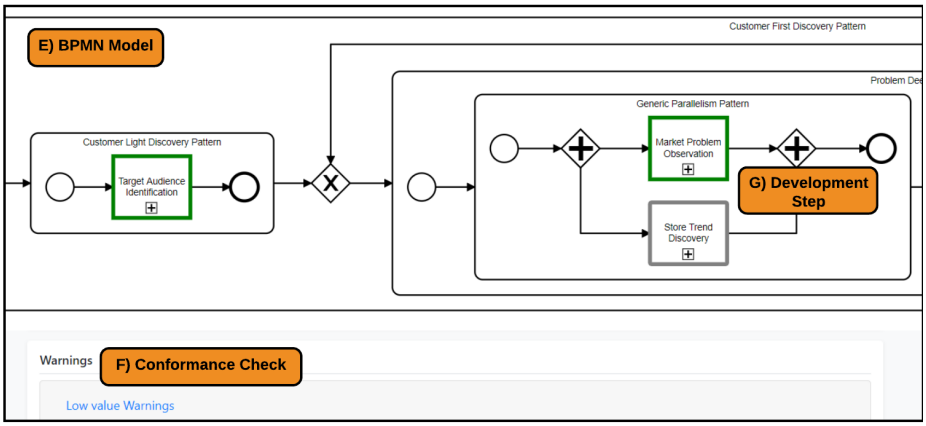

Figure 3. Composition of Development Method

3.3 Enactment of Development Method

Finally, the *Business Developer* needs to enact the composed development method. Here, the developer sees the BPMN model with completed, actual, and future development steps (e.g., Fig. 4) together with a corresponding Kanban board (see I). Based on that representation, he can do the suggested tasks. By starting the conduction, the business developer can read a description of the task and an overview of the artifacts that will be created and modified in those steps. Here, he can collaborate with all *Stakeholders* that are involved in the underlying method building block (see J). If one of the artifacts was linked to a canvas model with canvas building blocks during the composition, all *Stakeholders* could collaborate on the specific canvas with the support of knowledge of the linked canvas building blocks. For example, it is possible to visualize conflicts (e.g., if advertisement hurts privacy) between the placed elements on the canvas or compare the canvas with an existing company. Moreover, all created and modified artifacts in the development method can be accessed (see K).

4 Discussion and Outlook

In this demonstration paper, we have shown the software tool for our situation-specific business model development approach. Using DSR, we have conducted a literature review on business model development and a tool analysis of business model decision support systems to get aware of the problem. Out of that, we have developed the shown concept and implemented it in the *Situational Business Model Developer*. We evaluated our second design cycle by conducting a case study [18] on developing possible business models for a local event platform that will be created within a research project (see [12] for evaluation). We have created knowledge repositories for mobile apps based on a grey literature review (GLR) [8] and interviewed the project manager for the context of the project. We composed a development method with the phases of

discovery, analysis, design, development, and validation. We enacted the development method and received three potential business models of a content aggregator, a ticket seller, and a sponsored platform. By interviewing the project manager after conducting the case study, we concluded that guidance could support the development method by finding new activities (e.g., analyze store trends) and the business model itself by possible new decisions (e.g., lock-in pattern). Nevertheless, there are some limitations to our current approach. In the first stage, the GLR supports the provision of knowledge. Still, structuring method patterns with placeholders and arrangement of canvas building blocks with relationships are quite challenging. In the second stage, the context supports the recommendation of knowledge, but using nested method patterns and merging different canvas building blocks is less easy than expected. In the third stage, the enactment supports the traceability, but the reflection of context changes is challenging. In the future, we plan to improve our approach with a third design cycle by creating a modularized approach for situation-specific business model development. Here, we want to take the lessons learned from our tool analysis that currently every tool develops its decision support from scratch. Instead of that, we want to improve our approach so that each development step can be linked to a decision support module (e.g., cost calculator) and uses the artifacts that are defined in the step. Moreover, we aim to evaluate the cycle and the usability with a user study on different stakeholders.

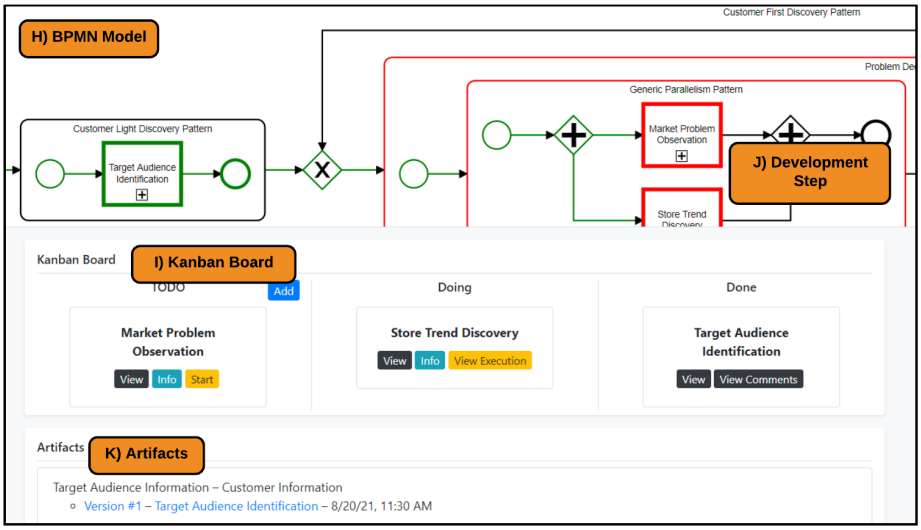

Figure 4. Enactment of Development Method

5 Acknowledgements

This work was partially supported by the German Research Foundation (DFG) within the CRC "On-The-Fly Computing" (CRC 901, Project Number: 160364472SFB901) and the German Federal Ministry of Education and Research (BMBF) through Software Campus grant (Project Number: 01IS17046).

References

1. Augenstein, D., Fleig, C.: Exploring Design Principles for a Business Model Mining Tool. In: Proceedings of the International Conference on Information Systems 2017. AIS (2017)
2. Bland, D.J., Osterwalder, A.: Testing business ideas. John Wiley & Sons, Hoboken (2020)
3. Boßelmann, S., Margaria, T.: Guided Business Modeling and Analysis for Business Professionals. In: Service Business Model Innovation in Healthcare and Hospital Management, pp. 195-211. Springer, Heidelberg (2017)
4. Bouwman, H., de Reuver, M., Heikkilä, M., Fielt, E.: Business model tooling: where research and practice meet. Electronic Markets 30(3), pp. 413-419 (2020)
5. Dellermann, D., Lipusch, N., Ebel, P., Leimeister, J.M.: Design principles for a hybrid intelligence decision support system for business model validation. Electronic Markets, 29(3), pp. 423-441 (2019)
6. CB Information Services: CB Insights 2019: Top 20 Reasons Why Startups Fail, https://www.cbinsights.com/research/startup-failure-reasons-top/
7. Frankenberger, K., Weiblen, T., Csik, M., Gassmann, O.: The 4I-framework of business model innovation: a structured view on process phases and challenges. International Journal of Product Development 18(3/4), pp. 249-273 (2013)
8. Gassmann, O., Frankenberger, K., Csik, M.: The business model navigator: 55 models that will revolutionise your business. Pearson, Harlow (2014)
9. Garousi, V., Felderer, M., Mäntylä, M. V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Information and Software Technology 100, pp. 101-121 (2021)
10. General Electric Inc: GE Global Innovation Barometer 2018, https://www.ge.com/reports/innovation-barometer-2018/
11. Gregor, S., Hevner, A.R.: Positioning and Presenting Design Science Research for Maximum Impact. MIS Quarterly 37(2), pp. 337-355 (2013)
12. Gottschalk, S., Yigitbas, E., Nowosad A.P., Engels, G.: Situation- and Domain-specific Composition and Enactment of Business Model Development Methods (To Appear). In: Product-Focused Software Process Improvement. Springer, Heidelberg (2021)
13. Hartmann, P.M., Zaki, M., Feldmann, N., Neely, A.: Capturing value from big data – a taxonomy of data-driven business models used by start-up firms. International Journal of Operations & Production Management 36(10), pp. 1382-1406 (2016)
14. Henderson-Sellers, B., Ralyté, J., Ågerfalk, P.J., Rossi, M.: Situational Method Engineering. Springer, Heidelberg (2014)
15. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly (28), pp. 75-105 (2004)
16. McGrath, R.G.: Business Models: A Discovery Driven Approach. Long Range Planning (43), pp. 247-261 (2010)
17. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken (2010)
18. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14(2), pp. 131-164 (2009)
19. Schoormann, T., Stadtländer, M., Knackstedt, R.: Designing business model development tools for sustainability—a design science study. Electronic Markets, pp.1-23. (2021)
20. Szopinski, D., Schoormann, T., John, T., Knackstedt, R., Kundisch, D.: Software tools for business model innovation: current state and future challenges. Electronic Markets 30, pp. 469-494 (2020)