



Situation- and Domain-Specific Composition and Enactment of Business Model Development Methods

Sebastian Gottschalk^(✉), Enes Yigitbas, Alexander Nowosad,
and Gregor Engels

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

Abstract. Developing effective business models is a complex process for a company where several tasks (e.g., conduct customer interviews) need to be accomplished, and decisions (e.g., advertisement as a revenue stream) must be made. Here, domain experts can guide the choices of tasks and decisions with their knowledge. Nevertheless, this knowledge needs to match the situation of the company (e.g., financial resources) and the application domain of the product/service (e.g., mobile app) to reduce the risk of developing ineffective business models with low market penetration. This is not covered by one-size-fits-all development methods without tailoring before the enactment. Therefore, we conduct a design science study to create a situation-specific development approach for business models. Based on situational method engineering and our previous work in storing knowledge of methods and models in distinct repositories, this paper shows the situation-specific composition and enactment of business model development methods. First, the method engineer composes the development method out of both repositories based on the situational context. Second, the business developer enacts the method and develops the business model. We implement the approach in a tool and evaluate it with a industrial case study on mobile apps.

Keywords: Business model development · Situational Method Engineering · Lean development · Kanban boards · Canvas models

1 Introduction

The development of effective business models is an important but also challenging task for companies to stay competitive. One reason for that is that customers want more and more integrated solutions for their perceived needs instead of

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).

single products [31]. Therefore, the business model can be even more important than the latest technology of the product [8]. Here, a study of CB Insights [7] in 2019 analyzed 101 bankrupt startups and concluded that 42% of them failed due to a missing market need. But also for established companies, the GE Innovation Barometer [14] in 2018 stated that 64% of the over 2000 business executives have the problem of developing effective business models for new ideas.

The development of business models is a complex and creative activity that consists of different phases (e.g., discover, develop, etc.) where multiple tasks (e.g., conduct customer interviews, analyze competitors, etc.) need to be accomplished [13]. Inside those tasks, communication and collaboration between different stakeholders (e.g., business developer, customer, etc.) often occur [10]. To support business model development, companies often rely on light visualization tools like kanban boards [18] for structuring the development process or canvas models [24] for structuring the information in different steps of the process. Due to the complexity of the process, there are several options for each process step, and as a consequence, a process with the wrong activities can lower the quality of the business model. Here, the guidance of a domain expert can support the development so that every stakeholder has the same needed understanding of the used methods on the kanban board and knowledge in the models [27]. In literature, different domain experts propose various methods to develop such business models in the form of development processes (e.g., [23]) and method repositories (e.g., [4]). Moreover, these experts provide knowledge in the form of taxonomies of possible (e.g., [19]) and patterns of successful (e.g., [12]) business models. However, the method should match the company's current situation (e.g., financial resources, target market size), and the information in the canvas models need to match the application domain (e.g., mobile app, social network) of the product/service of the company [28]. This, in turn, raises the chance of developing an effective business model for the company. Otherwise, the development of an ineffective business model can lead to poor market penetration of the product/service or even a company bankruptcy [26]. Although various business model development approaches have been proposed, they do not cover the step of tailoring the method to the current situation [17]. Tailoring by composing the method out of different method parts can include the situational context instead of a fixed one-size-fits-all development method for all contexts. Therefore our research question (RQ) is: *How to enable the situation-specific composition and enactment of business model development methods?*

To answer this question, we conduct a design science research (DSR) study [21] to develop an approach and a development tool. In our approach that supports non-experts in the development, the method engineer creates a method repository and models repository from the knowledge of different domain experts. We have already covered this in the past [16,17]. This paper aims to show the composition and enactment of the business model development method out of these repositories. While the method engineer should have high modeling capabilities based on Business Process Model and Notation (BPMN) for the method and feature models for the canvas during the composition, the business developer can stay with his lightweight structuring techniques of kanban boards and can-

vas models during the enactment. In comparison to existing methods for business model development, our approach focuses on the importance of the method engineer before the development. We implement the approach in an open-source tool and evaluate it based on a industrial case study of developing the business model for a local event platform. Our scientific contribution is the applied concept of situational method engineering to business model development while companies in practice are supported with a tool to develop their business models.

Concerning DSR, we structure the rest of the paper as follows: Sect. 2 covers the research background regarding business model development and situational method engineering. Section 3 provides insights into our DSR process. Section 4 shows the requirements of our solution divided into the provision of the method and knowledge repositories, the composition of the development method, and the enactment of the development method. Based on them, a concept is presented in Sect. 5 and implemented in Sect. 6. Section 7 evaluates the approach with a case study on developing a business model for a local event platform. Finally, Sect. 8 concludes the paper and gives an outlook on the next DSR cycle.

2 Background

2.1 Business Model Development

Business models can be defined as the rationale of how the organization creates, delivers, and captures value [24]. The development of business models is a complex task that often requires collaboration between different internal and external stakeholders [10]. To structure the complex process, some approaches like the BMI Magic Triangle [11] or the Cambridge Business Model Innovation Process [13] propose different phases (i.e., initiation, ideation, integration, implementation for [11]). Moreover, it is a crucial collaboration aspect to conduct experiments with the customers regularly to unfold their hidden needs [23]. Inside the different process activities, often light-weight visualization tools in the form of canvas models are used. Here, for example, the Value Proposition Canvas [25] summarizes the expected value proposition for a customer group, and the Business Model Canvas [24] visualizes the most important aspects of a business model. Moreover, the process can be supported with software-based Business model Development Tools (BMDTs). While the tools in practice mainly provide design support for business models based on the Business Model Canvas [30], there are also first approaches in research that integrate the knowledge of methods and models. For the methods, some approaches [9, 32] propose ideas for BMDTs that provide software support for different phases (i.e., analysis, design, implementation, and management in [9]). For the models, some approaches provide the usage of domain-specific knowledge [5] or the usage of patterns [22] for the development of the models. However, none of these tools deeply integrate both knowledge sources and method composition prior to the enactment.

2.2 Situational Method Engineering

Situational Method Engineering (SME), with its origin in software development, aims to create a development method based on the situation of a specific project

[20]. For that, SME has the role of a method engineer who analyzes various methods and stores them in a method repository. After identifying the context of the project, the engineer composes a situation-specific development method out of the method base. This development method, in turn, is then enacted by the project manager to run his project. To structure the method base, a method can be divided into method fragments that are reusable atomic blocks that have a process (called work unit), a product (called work product), or a producer focus [6]. These method fragments are combined to method components that transform inputs of work products into outputs of work products and are tailored into methods. Besides their origin in software development, some approaches also cover the business aspects of the projects. Here, some approaches [3, 15] cover business aspects as situational factors (i.e., customer, market characteristics, product characteristics, and stakeholder involvement in [3]) or use canvas models as work products (i.e., IoT Canvas in [15]). However, none of these approaches incorporates the whole development cycle of business models or uses an additional repository for the knowledge of the models.

3 Research Approach

This study uses design science research (DSR) to build an approach for the situation-specific development of business models. We use DSR because it focuses on the creation and incremental improvement of innovative artifacts based on existing theories. As method, we choose the DSR cycle of Kuchler and Vaishnavi [21] and based our research on the theories of opportunity creation [1] and boundary objects [29]. The opportunity creation theory states that businesses are co-created under high uncertainty [33]. Here, the development is an entrepreneurial process where companies create a business model based on their assumptions that need to be validated with the customers. Therefore, the process needs both parts of exploitation and exploration. The bounded object theory states the development is a heterogeneous task that requires the collaboration of different stakeholders with different knowledge [27]. Therefore, a common understanding between all stakeholders needs to be achieved. The process is shown in Fig. 1 and consists of two cycles with the five steps of taking *Awareness of [the] Problem*, making *Suggestion* for the solution, the *Development* of a corresponding artifact, the *Evaluation* of our solution, and the drawing of a *Conclusion*.

Based on both applied theories, in the *First Cycle*, we reviewed literature on business model development. Moreover, we conducted a systematic literature review on decision support systems for business model development. Based on that, we created conceptional parts for the situation-specific development of business models. For that, we used feature models to store the business model information of various business models. Out of this, a concrete business model for a single business model can be derived as an instance of the feature model. Moreover, we created a process to create and adapt those business models based on the conduction of experiments. Here, we evaluate the approach in a feasibility study with a tool implementation and the application of a usage scenario.

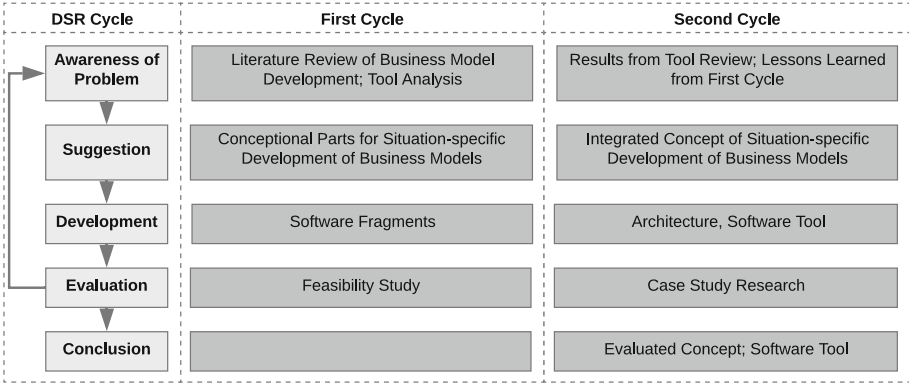


Fig. 1. Design science research process based on Kuechler and Vaishnavi [21]

In the *Second Cycle*, we took the lessons learned from the last cycle and the tool review to create an integrated concept of the approach. Here, we worked on the extensibility of the approach by concerning knowledge about methods and models from different domain experts. For that, we have used SME to derive a method repository with various methods to develop and validate business models for mobile applications [17]. Moreover, we have worked on an approach to consolidate the knowledge about business models from different real-world domain experts [16]. Based on both separate parts, we developed an integrated approach consisting of methods and models for the situation-specific development of business models. After implementing the tool and evaluating a case study on mobile apps, we concluded with an evaluated concept and a software tool.

4 Solution Requirements

At the beginning of our DSR study, we reviewed the literature regarding business model development and analyzed tools on decision support systems for business model development to get awareness of the challenges of software-based business model development. Out of this, we derived initial generic requirements that we refine to the current solution requirements. By considering the two stages of construction and development of a method from situational method engineering [20] and splitting up the provision from the method base from the construction of the method, we structure those requirements according to the topics of (R.1) Knowledge Provision of Methods and Models, (R.2) the Composition of the Development Method, and (R.3) the Enactment of the Development Method.

The requirement (R.1) *Knowledge Provision of Methods and Models* states that the solution should provide a variety of expert knowledge from which situation-specific development methods could be constructed. The usefulness of the approach profoundly relies on the usage of an appropriate method and corresponding models. Therefore, the solution needs a (R.1.1) *Storing of Expert Information* for different methods and models. Moreover, the approach depends on the

situation of the company and the application domain of the product/service. As a consequence, the solution needs a *(R.1.2) Characterisation of Context* both for the company and the product/service. Visualization can help to simplify the work with the knowledge. Therefore, a *(R.1.3) Visual Representation of Knowledge* both for the methods and models is needed. Last, business model development is a continuous process where different stakeholders are involved in different process activities. Therefore, the solution needs to cover an *(R.1.4) Understandability of Knowledge* around all stakeholders and an *(R.1.5) Extensibility of Knowledge* during the process.

The requirement *(R.2) Composition of Development Method* states that the development method should be composed out of the expert knowledge from the methods and models by taking the context of the company into account. The approach highly relies on the situational factors of the company and the application domain of the product. Therefore, the solution should cover the explicit *(R.2.1) Identification of Context* before the composition of the development method. The composition of the method could be based on a huge amount of knowledge in the form of methods and models. As a consequence, the solution should provide *(R.2.2) Assistance in Method Composition* based on the context. The development of business models is a process under high uncertainty so that not all choices can be covered in advance. Therefore, the solution should allow a *(R.2.3) Generalization of Method Composition* to provide different business model development processes simultaneously and an *(R.2.4) Adaptation of Method Composition* that provides a runtime adaptation to a changing context.

The requirement *(R.3) Enactment of Development Method* states the development method should be enacted so that business models could be developed on top of the knowledge. The development of business models is a complex task that should be supported with a software tool. Therefore, the solution should provide *(R.3.1) Executability of Method Enaction* to reduce the complexity. Business model development is a task with high uncertainties so that experts can not cover all knowledge in advance. As a consequence, the solution should cover *(R.3.2) Storing of Company Knowledge* so that the company can add internal methods and models. Moreover, the process is a complex task where different stakeholders are involved in different activities. Therefore, the solution should provide *(R.3.3) Traceability of Method Enaction* to reason all decisions in the past together with *(R.3.4) Stakeholder Involvement in Method Enaction* to allow the collaboration of different stakeholders in the activities.

5 Solution Concept

To address the solution requirements, we build our integrated approach for the situation-specific development of business models. An overview of the approach is shown in Fig. 2 which consists of the five roles of the *Meta-Method Engineer*, the *Method Engineer*, the *Domain Expert*, the *Business Developer*, and other *Stakeholders* together with the three stages of *(1) Knowledge Provision of Methods and Models*, *(2) Composition of Development Method* and *(3) Enactment of*

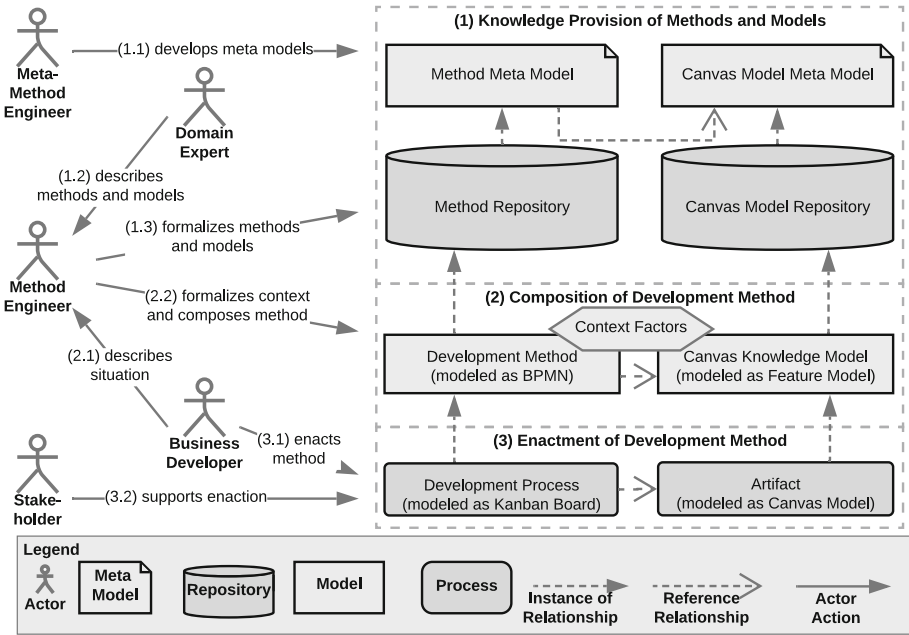


Fig. 2. Overview of the situation-specific business model development approach

Development Method. While we shortly describe each stage in the following, the respective subsections provide a more detailed explanation.

In the (1) *Provision of Methods and Knowledge Repository*, we provide general knowledge about methods to use and models to rely on within the business model development. For that, the *Meta-Method Engineer* needs first to create meta-models of how the methods and models should be structured (1.1). After that, different *Domain Experts* explain their knowledge about methods and models to the *Method Engineer* (1.2). The *Method Engineer*, in turn, formalizes the expert knowledge according to the meta-models to make them accessible during the composition of the method (1.3). In the (2) *Composition of Development Method*, the development method is composed out of both repositories. Here, the *Business Developer* explains the current context in which the business model should be developed to the *Method Engineer* (2.1). The *Method Engineer* formalizes this context as the situation of the method and the domain of the model. The engineer composes a situation-specific development method (2.2) consisting of the method itself as BPMN and the canvas knowledge models as feature models. In the (3) *Enactment of Development Method*, the composed method is enacted to develop the business model. Here, the *Business Developer* enacts the composed method (3.1) consisting of the development process as kanban board and the artifacts as canvas models. During this enaction, the development can be supported by other *Stakeholders* (3.2) (e.g., Designer).

5.1 Knowledge Provision of Methods and Models

The first stage, as shown in Fig. 3, aims to store all information of methods to use and knowledge to rely on from multiple *Domain Experts* in a structured format so that they can be used as resources during the development of business models. For that, the *Meta-Method Engineer* needs to create a *Method Meta Model* and a *Canvas Model Meta Model*. Here, we have already worked on modeling the methods [17] and the models [16] together with exemplary repositories in the past. Out of both repositories, we are now able to create the *Method Repository* and the *Canvas Model Repository*. Here, the *Method Engineer* formalizes both information from different *Domain Experts*.

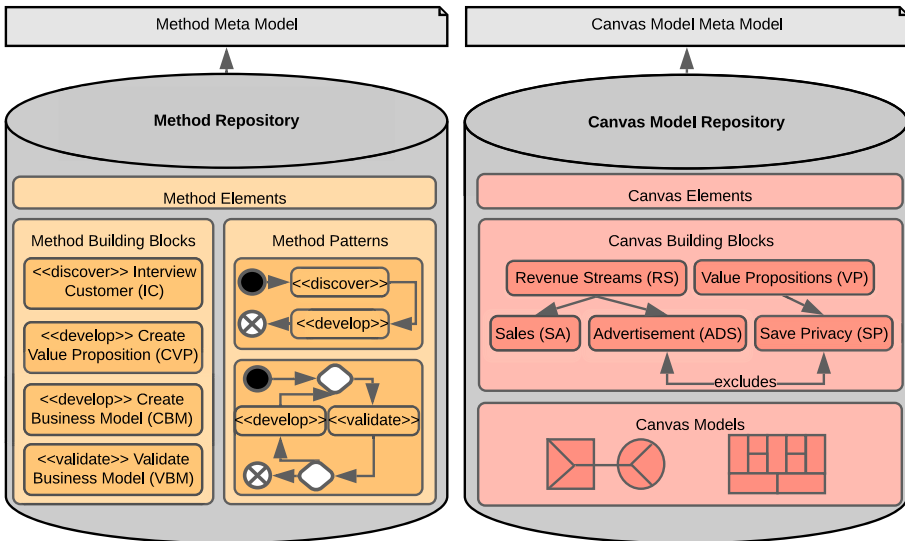


Fig. 3. Exemplary knowledge provision of methods and models

For that, we have developed repositories for the methods and the models (*R.1.1*). Inside the *Method Repository*, we adapt the concept of situational method engineering [20] and have the *Method Elements*, the *Method Building Blocks*, and the *Method Patterns*. Here, *Method Elements* are atomic parts of a method that can be divided into the possible situational factors, different types of methods, performed tasks, involved stakeholders, created artifacts, and used tools. These elements are combined to *Method Building Blocks*, where each block can have different situational factors, a task, different types, the involved stakeholders, and tools that can transform input artifacts into output artifacts. These building blocks are structured according to *Method Patterns*, which are BPMN Process Parts with situational factors when they should be used (*R.1.2*), and placeholders in which specific types of building blocks could be inserted. Inside the *Canvas Model Repository*, we adapt the concept of feature models [2]

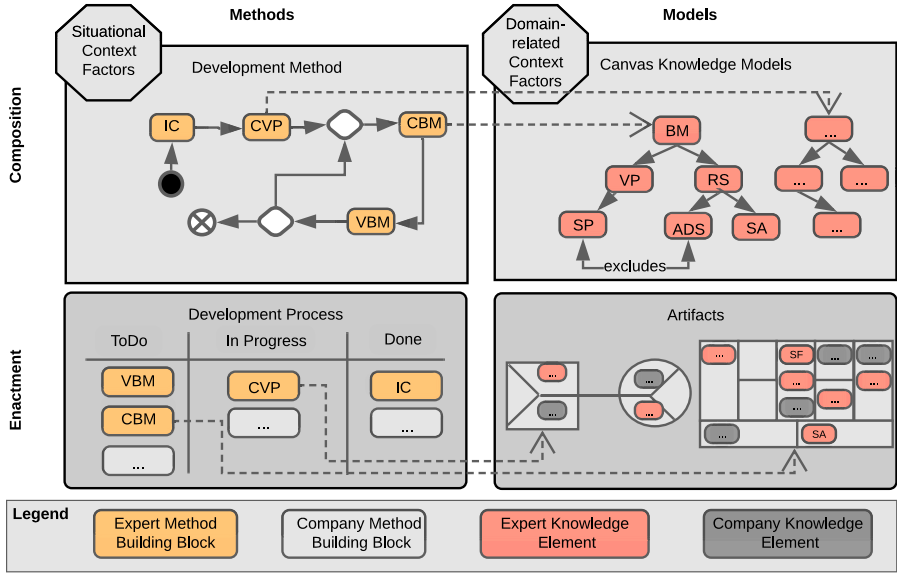


Fig. 4. Exemplary composition and enactment of development method

and have *Canvas Elements*, the *Canvas Building Blocks*, and the *Canvas Models*. Here, *Canvas Elements* are single chunks of knowledge that can be presented in a canvas model. Those elements are structured into a hierarchy within the *Canvas Building Blocks*. For the structuring, we use standard feature model relationships (e.g., requires, excludes) together with own relationships to save positive (e.g., supports) and negative (e.g., hurts) relationships between the elements. Moreover, we keep good practice patterns, and exemplary companies as instances of the building block together with an application domain of the block (R.1.2). Last, we provide *Canvas Models* (e.g., Value Proposition Canvas) as representations to visualize the building blocks. We ensure the understandability with descriptions for all important knowledge (R.1.3) and extensibility with the support for different experts and linking all information to specific experts (R.1.4).

5.2 Composition of Development Method

The second stage, as shown in Fig. 4, aims to use the context of the company to compose a business model development method. For that, the *Business Developer* describes the current context of the situation of the company and the application domain of the product/service to the *Method Engineer*. The *Method Engineer* formalizes this as situational factors of the *Method Repository* and a list of the domains from the *Canvas Model Repository* to compose the method.

The *Method Engineer* starts the composition of a BPMN process by choosing a *Method Pattern* from the *Method Repository* that is recommended to him by a matching of identified situational factors and the factors of the method (R.2.1).

After that, he can iteratively fill the placeholders in the patterns with *Method Building Blocks* or other patterns based on their type and recommendations by the factors. Moreover, he can check the conformance of the development process by finding wrong-filled placeholders or flows with input artifacts that have not been defined as outputs before. After that, the *Method Engineer* needs to connect the models from the *Canvas Model Repository* to the composing method. For that, he gets notified about *Method Building Blocks* that use *Canvas Models* as an output artifact. Here, he can select specific parts of *Canvas Building Blocks* as *Canvas Knowledge Models* for each canvas that is recommended to him based on matching the identified domain and the application domain of the building block. If he selected multiple building blocks for a single canvas, he needs to consolidate the knowledge as proposed by us in [16] (R.2.2). Moreover, at any time, he can create multiple development methods (R.2.3) and change the context factors and receives recommendations on how the method should be adapted (R.2.4).

5.3 Enactment of Development Method

The third stage, as shown in Fig. 4, aims to enact the composed method to allow the collaboration between different stakeholders during the development steps. For that, the *Business Developer* enacts the development methods in a lightweight process engine to receive an executable process.

The process engine is based on a *Kanban Board* where the development steps out of the composed method are grouped into *todo*, *in progress*, and *done steps* (see Fig. 4) (R.3.1). In every activity in progress, the *Business Developer* can communicate with all other *Stakeholders* that are mentioned in the definition of the corresponding *Method Building Block*. Moreover, if the building block is linked to a *Canvas Model*, the different stakeholders can collaborate on the specific canvas (R.3.4). Here, the knowledge from the connected building blocks can be used as recommendations. The whole process with every step is traceable for all stakeholders so that every decision that has been made can be reasoned over time (R.3.3). Moreover, the *Business Developer* can add his own method steps (e.g., a special type of interview) and canvas elements (e.g., a special type of advertisement) during the enaction to support flexible decision making (R.3.2). Out of this process, the business model is developed over time.

6 Solution Implementation

Based on the solution concept, we provide an implementation of the so-called *Situational Business Model Developer*. Our tool supports all three proposed stages, is released as open-source¹ and can also be directly used in the web browser². For that, the tool uses Angular to structure the web app, PouchDB to save all generated data in the web browser's storage, and BPMN.io to support the method representation. In the following, we explain the technical architecture and show the tool support.

¹ <https://github.com/SebastianGTTS/situational-business-model-developer>.

² <http://sebastiangtts.github.io/situational-business-model-developer/>.

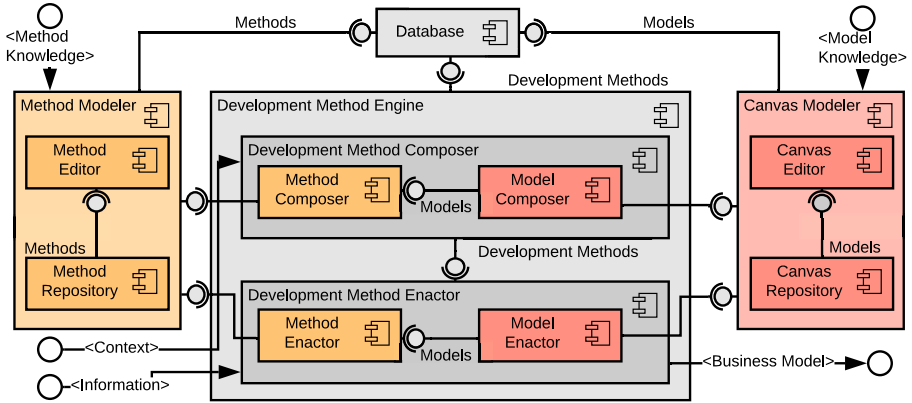


Fig. 5. Architecture of the *Situational Business Model Developer*

6.1 Architecture

The high-level architecture of our tool can be seen in Fig. 5. It consists of the *Database* of PouchDB to store the methods, models, and development methods together with the *Method Modeler*, the *Canvas Modeler*, and the *Development Method Engine*. The *Method Modeler* receives the *Method Knowledge* and stores it in the *Method Repository* by using the *Method Editor* and the BPMN.io Framework. The *Canvas Modeler* receives the *Model Knowledge* and stores it in the *Canvas Repository* by using the *Canvas Editor* and custom canvas boards. The *Development Method Engine* consists of the *Development Method Composer* and the *Development Method Enactor*. The *Development Method Composer* takes *Context* and composes a development method with the *Method Composer*, using BPMN structures from the *Method Modeler*, and the *Model Composer*, using consolidation and conflict detection algorithms from the *Canvas Modeler*. The *Development Method Enactor* takes *Information* about the development and enacts the method by using kanban boards in the *Method Enactor* and canvas models in the *Model Enactor* to output a *Business Model*.

6.2 Tool-Support

The screenshots of our tool can be seen in Fig. 6. While we have already covered the provision of knowledge of methods and models in previous work, we focus here on the composition and enactment of development methods. First, we have the composition of the development method, which is done by choosing the situational factors and modeling the BPMN process. Second, we have the linkage to the knowledge models, which is done by choosing the domain-related factors and merge the knowledge models. Third, we have the enactment of the development method based on the kanban board. Fourth, we have the usage of the linked models based on canvas model boards.

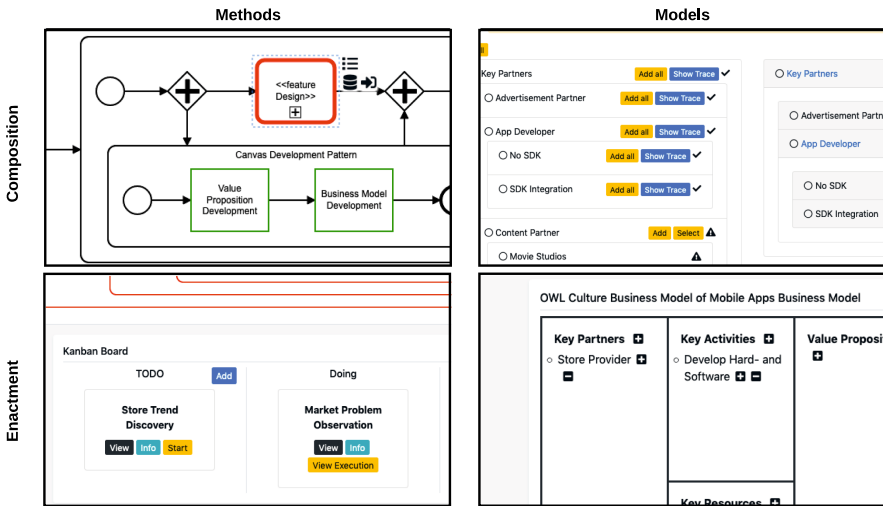


Fig. 6. Screenshots of the *Situational Business Model Developer*

7 Evaluation on Local Event Platform

To evaluate our approach, we conducted an industrial case study by developing a business model for a local event platform. For that, we explain our evaluation setting, provide execution of the study and analyze the results.

7.1 Evaluation Setting

Our aim is to investigate the integrated concept of situation-specific business model development to answer our stated research question. Here, we follow an explorative purpose to gather new insights for our third DSR cycle. For that, we provide the holistic case study of a single unit to develop a business model for OWL Live. The platform is created as part of an ongoing research project³ and acts as a two-sided platform between event providers and event visitors. Here, the owner wants to use new data mining techniques to aggregate events from different providers together with natural language processing to provide an enhanced recommendation system to the visitors. To gather the corresponding data, we combine the direct method of customer interviews with indirect methods of a grey literature review together with an analysis of existing information.

7.2 Execution of the Study

During the conduction, we structure our procedure according to the three stages of Knowledge Provision of Methods and Models, the Composition of the Development Method, and the Enactment of the Development Method.

³ Project Website: <https://www.sicp.de/en/projekte/owlkultur-plattform>.

The *Knowledge Provision of Methods and Models* has been made in previous research. Here, we have already created a method repository with various method elements, method building blocks, and method patterns based on a grey literature review on developing business models for mobile applications [17]. Moreover, for the canvas repository, we created existing canvas models of the Value Proposition Canvas (VPC) [25] and the Business Model Canvas (BMC) [24] together with a custom Feature Set Canvas (FSC) for storing possible features. While the knowledge of the BMC for the domain of mobile application, digital platforms, content aggregations, and social networks have been already created in [16], we created corresponding knowledge for the VPC and FSC.

In the *Composition of the Development Method*, we interviewed the project manager to gather the information about the current state of the platform (e.g., target customer), the situation of the project (e.g., `margSize:mass`), and the application domain of the app (e.g., content aggregation). This, in turn, allowed us to tailor a customized development method based on the phases of discovery, analysis, design, develop and validation of a method pattern in [17]. Out of the situational factors, we suggested identifying the target audience in the *discovery* phase, followed by a market problem observation to understand current customer pains and a store trend analysis to find trending features. Here, especially trend analysis is often missed by other approaches. In the end, the results for the event provider should be validated with customer interviews and the event visitors with a social media survey. In the *analysis*, we suggested running a market potential analysis together with a competitor analysis inside and outside the app stores. In the *design*, the value proposition, the business model, and the feature set need to be developed. Here, we linked the underlying canvas model to the specific canvas knowledge models inside our model repository and consolidated the specific knowledge based on the given application domain. Moreover, based on the models, a competitive advantage analysis and prioritization of the features should be done. In the *development* phase, we suggested the development of a beta-version in front of the product development. During development, the interest of the customers could be enhanced by using inbound marketing. Last, in the *validation*, we suggested the ongoing interview of both customer groups.

During the *Enactment of the Development Method*, the first target audiences of event providers and event visitors were identified and refined (e.g., culture actors for event providers, early adopters for event visitors) based on the prior feasibility study and the interview. For the *discovery*, the feasibility study also covered the observation of the market problem. At the same time, in the store trend analysis, possible features (e.g., invitation mechanism, social media connection) were outlined. The interview of customers and conduction of social media surveys are longer scheduled ongoing tasks. During the *analysis*, various statistics are looked up for the market potential. Moreover, existing knowledge [16] for local competitors and event apps are used for both competitor analyses. In the *design*, two value propositions (i.e., event providers, event visitors) were developed together with three possible business models (i.e., content aggregator, ticket seller, sponsored platform). Moreover, the feature sets for these business models were modeled (e.g., pipeline component for content aggrega-

tor), and the competitive advantage was analyzed (e.g., travel time calculator). This competitor analysis was also the foundation for feature prioritization. All of the canvases were structured with the tool and supported by the existing expert knowledge. Based on those structures, a competitor analysis was done. Currently, in the *development*, the beta version of the app is developed, while the ongoing customer interviews and social media surveys should validate the different business models and feature sets. Moreover, inbound marketing (e.g., landing page, social media posts) is planned to ensure high traffic during the upcoming beta. Because of the ongoing development, the continuous *validation* phase has not been started.

7.3 Analysis of Results and Implications

To evaluate our approach, we conducted a case study on developing a business model for OWL Live. Here, the development is an ongoing task for what we presented our results for the first four phases. By conducting the case study, we investigate that our approach supports the development method with guidance in new tasks to do (e.g., inbound marketing) and possible decisions to be made (e.g., lock-in mechanism). Moreover, the approach is generalizable to allow the development of different business models from the same knowledge and traceable to reason all changes over time. Nevertheless, we found some limitations during the composition and enactment that we want to discuss and fix in the next DSR cycle. We divide those limitations into the Restrictions of Expert Knowledge, Complexity of the Tool, and the Conduction of Single Case Study.

For the *Restrictions of Expert Knowledge*, we currently have just a base of knowledge for methods and models that focuses on mobile applications and, in particular, on event apps. This, in turn, limits the applicability of the approach in other scenarios. Therefore, we want to extend the knowledge and focus on models and methods for digital platforms in the future. For the *Complexity of the Tool*, we currently focused on the applicability of all provided features and dismissed a user experience that is easy to understand. This, in turn, limits the usage of the tool by end-users. Therefore, we want to increase the usability of the approach so that end-users can use it without prior introductions. For the *Conduction of Single Case Study*, we currently have applied the approach to the case of a local event platform. This, in turn, limits the information if the approach can be easily transferred to other scenarios. Therefore, we want to validate the transferability by creating several scenarios in a user study.

8 Conclusion and Future Work

The development of business models is a challenging task that can be supported by the knowledge of methods and models from different domain experts. Here, the knowledge needs to match the company's situation and the application domain of the product/service. Using two cycles of DSR, we have developed a situation-specific business model development approach. We implemented the approach in an open-source tool and evaluated it by conducting a case study

on a local event platform. Here, our results suggest that our approach supports business developers in developing business models by using the knowledge from existing methods and models. In the future, we will conduct a third DSR cycle to work on the extensibility of our approach and evaluate its usefulness in different scenarios. For that, we plan to modularize our concept so that single development steps (e.g., calculate business outcome) can be supported by different software modules. Moreover, we will evaluate our approach based on a user study in a lean development of mobile apps seminar where students have to develop business models for their apps over a more extended period.

References

1. Alvarez, S.A., Barney, J.B., Anderson, P.: Forming and exploiting opportunities: the implications of discovery and creation processes for entrepreneurial and organizational research. *Organ. Sci.* **24**(1), 301–317 (2013)
2. Apel, S., Batory, D., Kästner, C., Saake, G.: *Feature-Oriented Software Product Lines*. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-37521-7>
3. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The influence of situational factors in software product management: an empirical study. In: *International Workshop on Software Product Management*, pp. 41–48. IEEE (2008)
4. Bland, D.J., Osterwalder, A.: *Testing Business Ideas*. Wiley, Hoboken (2020)
5. Boßelmann, S., Margaria, T.: Guided business modeling and analysis for business professionals. In: Pfannstiel, M.A., Rasche, C. (eds.) *Service Business Model Innovation in Healthcare and Hospital Management*, pp. 195–211. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-46412-1_11
6. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. *Inf. Softw. Technol.* **38**(4), 275–280 (1996)
7. CB Information Services: CB Insights 2019: Top 20 Reasons Why Startups Fail. <https://www.cbinsights.com/research/startup-failure-reasons-top/>
8. Chesbrough, H.: Business model innovation: opportunities and barriers. *Long Range Plan.* **43**(2–3), 354–363 (2010)
9. Ebel, P., Bretschneider, U., Leimeister, J.M.: Leveraging virtual business model innovation: a framework for designing business model development tools. *Inf. Syst. J.* **26**(5), 519–550 (2016)
10. Eppler, M.J., Hoffmann, F., Bresciani, S.: New business models through collaborative idea generation. *Int. J. Innov. Manag.* **15**(06), 1323–1341 (2011)
11. Frankenberger, K., Weiblen, T., Csik, M., Gassmann, O.: The 4I-framework of business model innovation: a structured view on process phases and challenges. *Int. J. Prod. Dev.* **18**(3/4), 249 (2013)
12. Gassmann, O., Frankenberger, K., Csik, M.: *The Business Model Navigator: 55 Models that Will Revolutionise Your Business*. Pearson, Harlow (2014)
13. Geissdoerfer, M., Savaget, P., Evans, S.: The Cambridge business model innovation process. *Procedia Manuf.* **8**, 262–269 (2017)
14. General Electric Inc.: GE Global Innovation Barometer 2018. <https://www.ge.com/reports/innovation-barometer-2018/>
15. Giray, G., Tekinerdogan, B.: Situational method engineering for constructing internet of things development methods. In: Shishkov, B. (ed.) *BMSD 2018*. LNBP, vol. 319, pp. 221–239. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94214-8_14

16. Gottschalk, S., Kirchhoff, J., Engels, G.: Extending business model development tools with consolidated expert knowledge. In: Shishkov, B. (ed.) BMSD 2021. LNBI, vol. 422, pp. 3–21. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79976-2_1
17. Gottschalk, S., Yigitbas, E., Nowosad, A., Engels, G.: Situation-specific business model development methods for mobile app developers. In: Augusto, A., Gill, A., Nurcan, S., Reinhartz-Berger, I., Schmidt, R., Zdravkovic, J. (eds.) BPMDS/EMMSAD -2021. LNBI, vol. 421, pp. 262–276. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-79186-5_17
18. Gross, J., Mcinnis, K.: *Kanban Made Simple: Demystifying and Applying Toyota's Legendary Manufacturing Process*. AMACOM, New York (2003)
19. 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. *Int. J. Oper. Prod. Manag.* **36**(10), 1382–1406 (2016)
20. Henderson-Sellers, B., Ralyté, J., Ågerfalk, P.J., Rossi, M.: *Situational Method Engineering*. Springer, Heidelberg (2014). <https://doi.org/10.1007/978-3-642-41467-1>
21. Kuechler, B., Vaishnavi, V.: On theory development in design science research: anatomy of a research project. *Eur. J. Inf. Syst.* **17**(5), 489–504 (2008)
22. Lüdeke-Freund, F., Bohnsack, R., Breuer, H., Massa, L.: Research on sustainable business model patterns: status quo, methodological issues, and a research agenda. In: Aagaard, A. (ed.) *Sustainable Business Models*. PSSBIAFE, pp. 25–60. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-93275-0_2
23. McGrath, R.G.: Business models: a discovery driven approach. *Long Range Plan.* **43**, 247–261 (2010)
24. Osterwalder, A., Pigneur, Y.: *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. Wiley, Hoboken (2010)
25. Osterwalder, A., Pigneur, Y., Bernarda, G., Smith, A., Papadacos, P.: *Value Proposition Design: How to Create Products and Services Customers Want*. Get Started With. Strategyzer Series. Wiley, Hoboken (2014)
26. Ries, E.: *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. Crown Business, New York (2014)
27. Schwarz, J.S., Legner, C.: Business model tools at the boundary: exploring communities of practice and knowledge boundaries in business model innovation. *Electron. Mark.* **30**(3), 421–445 (2020)
28. Sosna, M., Treviño-Rodríguez, R.N., Velamuri, S.R.: Business model innovation through trial-and-error learning. *Long Range Plan.* **43**(2–3), 383–407 (2010)
29. Star, S.L., Griesemer, J.R.: Institutional ecology, 'translations' and boundary objects: amateurs and professionals in Berkeley's museum of vertebrate Zoology, 1907–39. *Soc. Stud. Sci.* **19**(3), 387–420 (1989)
30. Szopinski, D., Schoormann, T., John, T., Knackstedt, R., Kundisch, D.: Software tools for business model innovation: current state and future challenges. *Electron. Mark.* **60**(11), 2794 (2019)
31. Teece, D.J.: Business models, business strategy and innovation. *Long Range Plan.* **43**(2–3), 172–194 (2010)
32. Terrenghi, N., Schwarz, J., Legner, C., Eisert, U.: Business model management: current practices, required activities and IT support. In: *Proceedings of the Conference on Wirtschaftsinformatik*, pp. 972–986 (2017)
33. Vogel, P.: From venture idea to venture opportunity. *Entrep. Theory Pract.* **41**(6), 943–971 (2017)