

Don't Start from Scratch: A Modularized Architecture for Business Model Development Tools

Sebastian Gottschalk^{1,*}, Enes Yigitbas¹, Alexander Nowosad¹ and Gregor Engels¹

¹Software Innovation Lab, Paderborn University, Paderborn, Germany

Abstract

Software-based Business Model Development Tools (BMDTs) are developed in research and practice to support business model development (BMD). However, current BMDTs are mainly developed from scratch, which binds resources to basic feature development (e.g., discussion features) instead of actual support for the BMD (e.g., simulating business models). Therefore, we have developed an open-source modularized architecture that provides those basic features and can be used by BMDTs developers to integrate their development support in the form of software modules (e.g., business simulation module). The architecture is derived as a situated implementation of a conducted design study.

Keywords

Business Model Development, Software Modules, Software Tool

1. Introduction

The development of new and improvement of existing business models, defined by Osterwalder et al. as "the rationale of how the organization creates, delivers, and captures value", is a challenging topic for every organization [1]. Here, the development method for such business models consists of an iterative non-linear process where different steps have to be conducted (e.g., analysis of competitors) and decisions must be made (e.g., choosing a revenue model). That development, in turn, can be supported by BMDTs. Different BMDTs have been developed in research and practice in recent years. While those tools in practice focus on visualization (e.g., as canvas boards) and collaboration (e.g., on discussion boards) aspects [2], tools in research focus on providing support for the design (e.g., usage of business model patterns) and the decision-making (e.g., crowd validation) [3]. However, here, most BMDTs are developed from scratch, which binds the development resources to common functionalities (e.g., artifact management, collaboration) instead of novel techniques for supporting BMD (e.g., creativity support, scenario planning). Moreover, the developed artifacts of different BMDTs can not be automatically used by other BMDTs (e.g., one tool for the design and another for the validation of a business model). Therefore, we developed a modularized architecture to integrate different support techniques for BMD and the exchange of their artifacts in a single BMDT.

ICSOB '22: 13th International Conference on Software Business, November 8–11, 2022, Bolzano, Italy


*This work was supported by the DFG within the CRC "On-The-Fly Computing" (CRC 901, Project Number: 160364472SFB901) and the BMBF through Software Campus grant (Project Number: 01IS17046).

*Corresponding author.

✉ sebastian.gottschalk@uni-paderborn.de (S. Gottschalk); enes.yigitbas@uni-paderborn.de (E. Yigitbas); anowosad@mail.uni-paderborn.de (A. Nowosad); gregor.engels@uni-paderborn.de (G. Engels)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

In the past, we have already conducted a design science study to provide a solution for situation-specific business model development with three stages [4]. In the first stage, we utilize knowledge of different methods and models and store them in repositories. Next, the second stage provides guidance to compose a developed method out of those repositories for a specific situation. Last, the third stage enacts the development steps of the composed method to create and modify different artifacts, including the business model. Moreover, we released the BMDT called *Situational Business Model Developer (SBMD)* under open source [5].

Based on our solution, we developed a modularized architecture that supports the integration of various visualization, design, and decision support from different developers. For that, we extend our existing SBMD that can be directly used within the web browser¹ but which source code can also be downloaded². For the integration of different development support, we use modularization, which means splitting up the different software functionalities into separate parts so that they can be used and modified independently of each other [6]. Here, the specific development support will be designed as dedicated software modules containing development steps to conduct and artifacts to create. This, in turn, has the advantage for developers to concentrate on the novel support part for BMD and for the users to benefit from various development support simultaneously. As we based our modularized architecture on the Angular framework, we require the development support implemented as Angular modules. In the following, we explain the creation of modules (Sect. 2), their integration into the architecture (Sect. 3), and an outlook for future research (Sect. 4).

2. Creation of Modules

For the modularization, we split up every *Development Support Module* into different *Development Support Tools* and *Development Support Meta Artifacts*, as shown in Fig. 1. The support tools ensure the actual functionalities for the development support by providing different atomic *Development Support Steps* that are combined together for specific development steps of the development method. The meta artifacts specify the different *Support Artifacts* that are created and modified within the steps. By splitting the modules into tools and meta artifacts, we ensure the easy usage of artifacts by different tools together with their exchange.

For example, we created a *Canvas Module*, consisting of a *Canvas Tool* and a *Canvas Meta Artifact*. The tool has different steps for creating, updating, or comparing a canvas. For that, the meta artifact provides a structure to build different canvas modes like the Value Proposition Canvas or the Business Model Canvas. Those modules, in turn, need to be integrated into the architecture to make them usable.

3. Integration of Modules in Architecture

To integrate those modules into our modularized architecture, we provide interfaces and hooks for all the stages of our solution. While interfaces are used to receive information about the modules, hooks are used to extend the SBMD with new functionalities. For the first stage

¹Online Version of the Tool: <http://sebastiangtts.github.io/situational-business-model-developer/>

²Source Code of the Tool: <https://github.com/sebastiangtts/situational-business-model-developer>

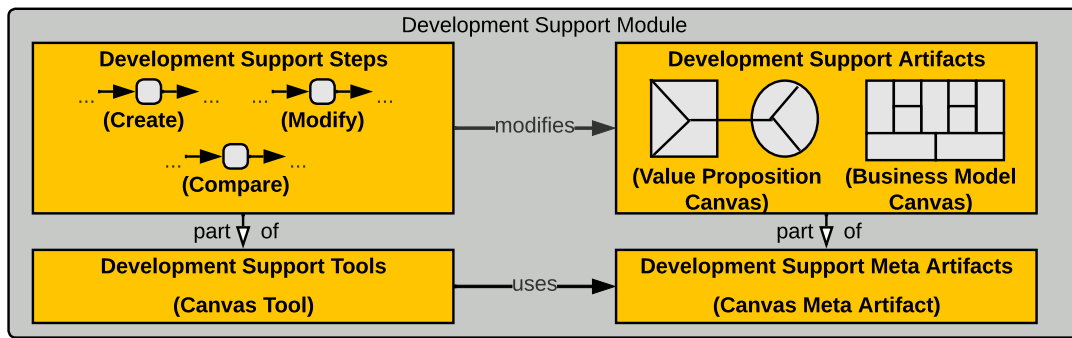


Figure 1: Exemplary Creation of a Development Support Modules

of knowledge provision, we provide interfaces to gather information about the tools, steps, meta artifacts, and artifacts, together with a hook to create predefined artifacts out of the meta artifacts. For the development method composition, we provide hooks to gather information and configure the single development steps during the composition. For the development method enactment, we provide hooks to execute the single steps to create and modify the artifacts. By integrating those hooks and interfaces, the development support can be added to specific development steps in the repository for the methods and, therefore, used within the enactment to support the business model development. For creating the modules, we provide a code generation based on Angular Schematics and additional documentation for the interfaces and hooks. Both are available in the repository of the source code.

An overview of how those *Development Support Modules* are used for the *Canvas Module* is shown in Fig. 2. The *Modularization of Development Support* consists of the *Development Support Meta Artifacts* and the *Development Support Tools*. For that, the *Development Support Meta Artifacts* provide information on how the artifacts are structured to allow their usage by different tools (e.g., *Canvas Meta Artifact*). The *Support Tools* are the concrete software parts that provide interfaces and hooks to allow development support and have references to the *Meta Artifacts*. Here, the *Tools* (e.g., *CanvasTool*) are accessed through atomic *Interface* (e.g., *createCanvas*, *createPrototype*), which allow flexible usage of the underlying functionalities. The *Application to Development Steps* consists of the *Artifacts* and the *Development Process*. For that, the *Artifacts* (e.g., *Business Model Canvas*) are created and modified during the development support. The *Development Process* contains different *Development Steps* (e.g., *Create Business Model*) with a stepwise usage of the development support. Here, those steps (e.g., 1, 2, 3 of *Create Business Model*) provide references to the corresponding interfaces (e.g., *createCanvas*, *editCanvas*, *viewCanvas* of *Canvas Tool*).

4. Conclusion and Outlook

Nowadays, existing BMDTs are mainly built from scratch, focusing on basic features instead of novel development support. To solve this challenge, we present a modularized architecture where development support can be added as software modules. Currently, we are working on a

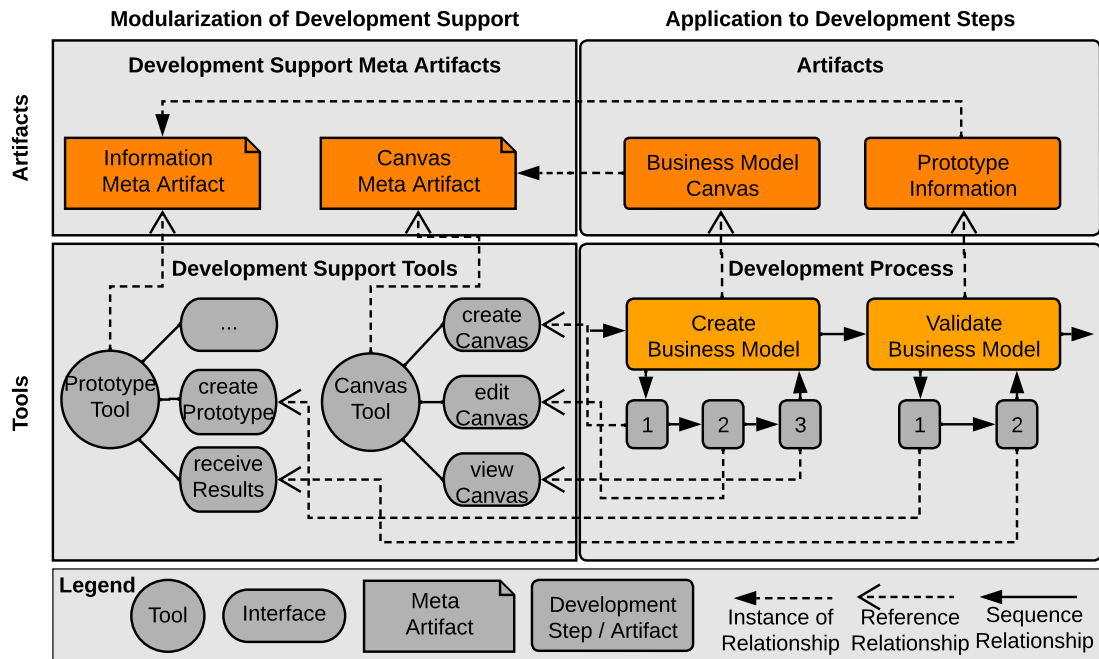


Figure 2: Exemplary Usage of Development Support Modules

new layout and improved documentation to improve the SBMD. Based on that architecture, we are, in parallel, implementing different software modules. With both, we want to encourage researchers and practitioners to use our tool for developing BMs and BMDTs.

References

- [1] A. Osterwalder, Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*, John Wiley & Sons, Hoboken, 2010.
- [2] D. Szopinski, T. Schoormann, T. John, R. Knackstedt, D. Kundisch, Software tools for business model innovation: current state and future challenges, *Electronic Markets* 60 (2019) 2794.
- [3] H. Bouwman, M. de Reuver, M. Heikkilä, E. Fiel, Business model tooling: where research and practice meet, *Electronic Markets* 30 (2020) 413–419.
- [4] S. Gottschalk, E. Yigitbas, A. Nowosad, G. Engels, Continuous Situation-specific Development of Business Models: Knowledge Provision, Method Composition, Method Enactment, in: *Journal of Software and Systems*, Springer, 2022.
- [5] S. Gottschalk, E. Yigitbas, A. Nowosad, G. Engels, Situational Business Model Developer: A Tool-support for Situation-specific Business Model Development, in: *Proceedings of the Wirtschaftsinformatik, AIS, Nuremberg*, 2022.
- [6] D. Parnas, On the criteria to be used in decomposing systems into modules, *Communications of the ACM* 15 (1972) 1053–1058.