Extending Business Model Development Tools with Consolidated Expert Knowledge*

Sebastian Gottschalk, Jonas Kirchhoff, and Gregor Engels

Software Innovation Lab, Paderborn University, Germany
{sebastian.gottschalk,jonas.kirchhoff,gregor.engels}@uni-paderborn.de

Abstract. Business Model Innovation (BMI) is a creative process that often needs collaboration between different stakeholders with the support of domain experts. Instead of innovation workshops where the domain experts need to be physically present, software-based tools allow reusing the knowledge of many domain experts independent of their actual presence. This reusing of expert knowledge, which improves the quality of the developed business models, is currently not supported by existing Business Model Development Tools (BMDTs). To address this shortcoming, we present an approach to support BMDTs with consolidated knowledge of different experts. In our approach, domain experts formalize their knowledge about business models for particular domains in expert models to make them useable within and transferable between different tools. Business developers can subsequently choose the expert models they need, consolidate the knowledge, and use it within the BMI process. With this approach, we provide a three-fold contribution to the research of BMDTs: First, we design a modeling language to store the business model knowledge of individual experts. Second, we develop a concept to consolidate expert knowledge and detect possible knowledge conflicts. Third, we provide blueprints to add expert knowledge into existing BMDTs. We demonstrate the technical feasibility of our approach with an open-source BMDT implementation and show the applicability with an exemplary instantiation of a local event platform.

Keywords: Business Domain Knowledge · Business Model Development Tool · Expert Knowledge · Business Model Innovation

1 Introduction

An essential task for a company to stay competitive is the continuous innovation of its business models, defined by Osterwalder et al. as "the rationale of how the organization creates, delivers, and captures value" [27]. The high complexity of this task is also one of the results of the GE Innovation Barometer 2018 [15],

* 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 European Regional Development Fund (ERDF, Funding Code: EFRE-0801186)

Preprint, cite this paper as:
Gottschalk S., Kirchhoff J., Engels G. (2021) Extending Business Model Development Tools with Consolidated Expert Knowledge. In: Shishkov B. (eds) Business Modeling and Software Design. BMSD 2021. Lecture Notes in Business Information Processing, vol 422. Springer, Cham. https://doi.org/10.1007/978-3-030-79976-2_1

The final authentificated version is available online at:
https://doi.org/10.1007/978-3-030-79976-2_1

a study with over 2000 business executives, in which 64% of these executives have mentioned the "difficulty to define an effective business model to support new ideas and make them profitable" [15]. By comparing the results with a previous study of 2015, the challenge is getting even larger (59% of over 3000 executives). An important reason for this is that customers expect solutions for perceived needs rather than just products [34]. These perceived needs result in the business model potentially being more important than the latest technology of the product [6].

One challenge in Business Model Innovation (BMI) is that the process of BMI is a creative task that often requires the collaboration of different internal and external stakeholders [11]. One group of these stakeholders are so-called domain experts who provide deep knowledge in a particular domain. Instead of collaborating directly with these experts, it is also possible to use their expert knowl-

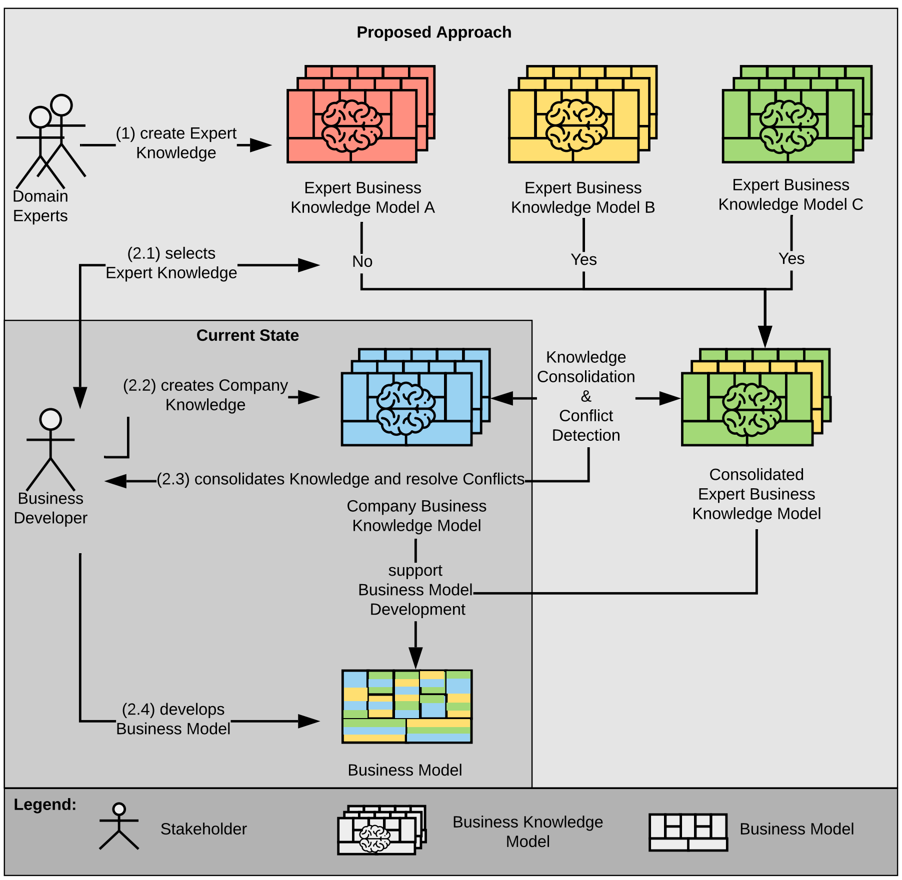


Fig. 1. A *Business Developer* can develop better business models by building upon existing domain knowledge provided by multiple *Domain Experts*

edge in the form of business model taxonomies (e.g., [23, 19]) or business model patterns (e.g., [13, 31]). Advantages of this expert knowledge are its cost-effect reusability independent from the actual presence of the expert. Consequently, software-based Business Model Development Tools (BMDTs) and the business developer as users can benefit from this expert knowledge to innovate their business models. Nevertheless, this reusing of expert knowledge is not covered by existing BMDTs in practice [33], and their underlying modeling languages [20].

In this paper, we present an approach that consolidates the knowledge of different experts to support business model development (see Fig. 1). For this purpose, we provide a modeling structure based on the concept of feature models [3], and the Business Model Canvas [27] where *Domain Experts* can store their knowledge about different business domains as shown in Fig. 1 *(1)*. The *Business Developer* selects the expert knowledge *(2.1)* he wants to use for innovating his business model. Moreover, he captures the business domain knowledge of the company *(2.2)*. Because the experts and the company may use different vocabulary and contrary ideas, the knowledge of the experts needs to be consolidated, and conflicts in the knowledge between the experts and the company need to be resolved. For this, we present a concept to consolidate expert and company knowledge and detect conflicts. Out of this consolidating process, the *Business Developer* receives a homogeneous knowledge base *(2.3)* with all knowledge relevant to him. This homogeneous knowledge base, in turn, will support him in developing new business models for his company *(2.4)*. This can be done by discovering business elements, suggesting business patterns, and comparing business models.

Our approach provides a threefold contribution to the research of software-based business model development. First, we provide a ready-to-use modeling language for expert knowledge that can be implemented and used in existing tools. Second, we develop concepts for the consolidation of different expert knowledge and the handling of conflicts between them. Third, we provide blueprints on how expert knowledge can support the process of business model development in BMDTs. Moreover, we implement our concept in an open-source BMDT and apply it with an exemplary instantiation for the development of a business model of a local event platform.

The rest of the paper is structured as follows: Section 2 provides the background in terms of business model development and feature models. Section 3 explains the solution concept for the modeling language, the concept, and the blueprints. Their technical implementations are shown in Sect. 4. The application of the approach is shown in Sect. 5. Section 6 presents the related work of our approach. Finally, we conclude our paper in Sect. 7.

2 Background

In this section, we show the background of our work which can be divided into the process of business model development (Sect. 2.1) and the usage of feature models (Sect. 2.2).

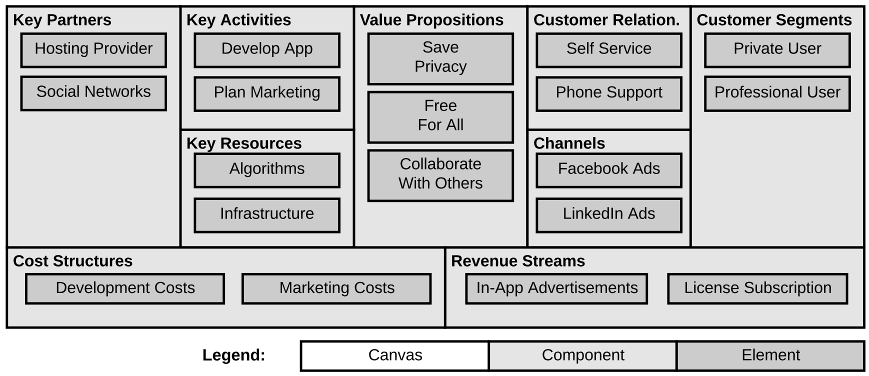


Fig. 2. Structure of the Business Model Canvas with *Components* and *Elements* based on the example of a mobile to-do app

2.1 Business Model Development

The process of business model development is a creative task that often requires creativity and collaboration between different stakeholders [11], together with a deep analysis of the market, existing competitors, and potential customers [34]. A common setting to develop new business models are workshops [14]. In these workshops, different stakeholders try to understand the current needs of the customers and develop possible solutions, often with the help of expert knowledge like patterns [13] or taxonomies [22].

In these workshops, the structuring of insights can be supported by business model modeling languages (BMMLs) like the e3-Value Model [16] or the Business Model Canvas (BMC) [27]. While many languages have been developed over the years [20], the BMC [27] is the de-facto standard for business modeling. The BMC divides the business model into the nine components of *Customer Segments*, *Value Propositions*, *Channels*, *Customer Relationships*, *Key Activities*, *Key Resources*, *Key Partners*, *Revenue Streams*, and *Cost Structure*. An example of the BMC for the business model of a to-do app can be seen in Fig. 2. The example consists of different customer segments (e.g., *Private User*) from which money can be generated through different revenue streams (e.g., *In-App Advertisements*). While, in practice, the structuring of different elements in a single canvas is done with different colored sticky notes [12], the underlying work [26] also introduced a Business Model Ontology (BMO) for formalizing the relationships between the different components. This, in turn, can be used to understand the dependencies between the modeled elements. This ontology can be directly used in an editor [12] but is also the basis for the concepts of dynamic business models [8] and meta-modeling [24]. To cover the maturity of the different BMMLs, Alberts et al. [1] present a meta-model for BMMLs based on the Meta-Object Facility (MOF). Moreover, to support the modeling and comparison of different business models, Osterwalder et al. [29] provide the idea to model different types of business models as taxonomies so that concrete businesses can be interpreted as instances of these taxonomies. These taxonomies can also be represented through feature models [17]. Moreover, the business model development can be supported by software-based tools.

These software-based tools are often called Business Model Development Tools (BMDT) and provide different guidance levels to develop new and improve existing business models [33]. Here, earlier examples of these tools in the literature focus on the visualization of the business model [12] or simple financial assessments [16]. An analysis of business modeling tools in practice [33] shows that those tools focus on the design of business modeling but not on the actual decision support. Nevertheless, a shift from simple design support of business modeling to real decision support by these tools needs to be done [28].

2.2 Feature Models

The concept of feature models is part of Software Product Lines (SPLs) that can be defined as "a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" [7]. Here, feature models are used to structure this common, managed set of features in a hierarchical model. An example for a feature model, which we applied to the business modeling in [17], can be seen in Fig. 3. Here, the hierarchy refines the top feature of the *Canvas* (e.g., *Business Model Canvas*) into the sub-features of the *Components* (e.g., *Customer Segments*). Next, these features are refined to *Elements* (e.g., *Private User*) and could be further refined to sub-elements.

Features can be *Mandatory* (e.g., *Value Propositions*) or *Optional* (e.g., *Customer Segments*) for the model instances. Moreover, there can be *Or* (at least one sub-feature is selected / e.g., *Save Privacy* or *Collaborate with Others*), and *Xor* (exactly one sub-feature is selected / e.g., *Private User* xor *Professional*

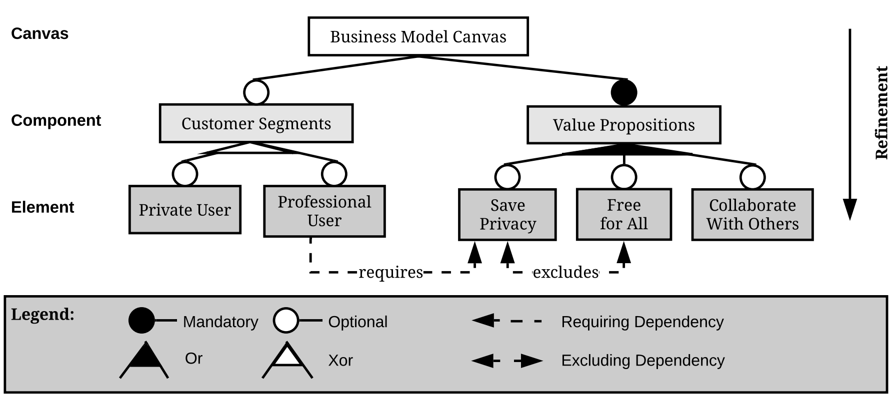


Fig. 3. Structure of feature models with an refinement to the *Components* and *Elements* of the Business Model Canvas

User) relationships between a parent and a child feature. To refine the model instance, cross-tree constraints for *requiring* (e.g., *Professional User* requires to *Save Privacy*) and *excluding* (e.g., *Save Privacy* is excluded from *Free for All*) dependencies can be made. A big issue in SPL development, which also exists in modeling the expert knowledge of business models, is to find the right granularity for the features [21].

3 Solution Concept

In this section, we describe the solution concept to add the support of consolidated expert knowledge to Business Model Development Tools. For that, we first define a modeling language to store expert knowledge (Sect. 3.1). Based on that, we introduce concepts for knowledge consolidation and conflict detection (Sect. 3.2) together with blueprints on how expert knowledge can be used in BMDTs to support the development process (Sect. 3.3).

3.1 Modeling of Expert Knowledge

To allow the consolidation of expert knowledge, the *Domain Experts* need to store their knowledge into distinct *Expert Business Knowledge Models*. For this, we use the concept of feature models [3] that we already transferred to business modeling in the past [17]. The structure of these business models, based on the Business Model Canvas [27], can be seen in Fig. 3. While these models can cover the basic information of the business models, we need to cover additional information from the domain expert to allow a reusing of the knowledge. These additional information include knowledge about the model itself, the meaning of the possible features and the relationships between different business model elements. Moreover, we want to store possible instance sets of the features that can be either the elements for an exemplary company or patterns used in successful business models.

The meta-model for storing expert knowledge can be seen in Fig. 4. It consists of all constraints and relationships which are previously shown in Fig. 3. Moreover, we add additional information about the *FeatureModel* itself (*name*, *description*, *version*, *copyright*) and the *Author* (*name*, *company*, *email*, *website*) to give the *Business Developer* initial information about the *Domain Expert* and the application domain of the model. Additionally, we add a *description* to the *Feature* to provide a uniform understanding of the feature between the *Domain Expert* and all *Business Developer* who use the model. In addition to the hard *CrossTreeRelationships* of *Requires* and *Excludes*, we add some softer constraints in the form of *Supports* and *Hurts* as relationships between the features. These softer constraints, which are also used in requirements engineering [35], can be used by the *Domain Expert* to model recommendations between the elements (e.g., if the *Business Developer* considers this feature he should / should not also consider this feature). Moreover, we explicitly model sets of these features as *Instances*. Here, *Patterns* are describe good combination of features (e.g.,

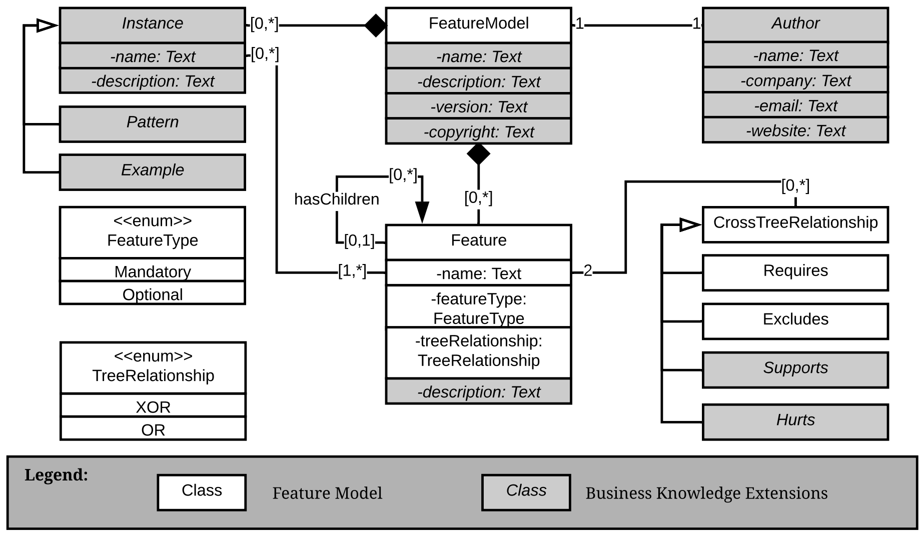


Fig. 4. Meta-Model of the *Business Knowledge Model* which is based on feature models and is extended with additional information

combining *Freemium* and *Mass-Market* for a gaming app) and *Examples* describe used combinations by existing companies (e.g., features of the business model of a particular gaming app). For both of them, we add a *name* and an additional *description* to ensure an unified understanding.

3.2 Knowledge Consolidation and Conflict Detection

After the *Domain Experts* have stored their knowledge into the *Expert Business Knowledge Models*, the knowledge needs to be consolidated so that it can be used by the *Business Developer*. For that, we are using the nine components (e.g., *Value Propositions*, *Customer Segments*) of the Business Model Canvas [27] as a starting point to merge the different knowledge models that the *Business Developer* wants to use. From this point, we provide the *Business Developer* assistance in merging the elements of the business knowledge model with the expert knowledge models into the homogeneous knowledge base. Here, the developer can add new elements of the expert models, merge elements with the same namings, and merge elements with different naming. For both of the mergings, merging conflicts between the different models can occur. These conflicts need to be detected so that the *Business Developer* can resolve them.

To detect conflicts, we analyze the model in the merging process in terms of the conflicts mentioned in Table 1. We divide the conflicts into the three categories of *Feature Types*, *Tree Relationships*, and *Cross-Tree Relationships*. The conflicts in *Feature Types* and *Tree Relationships* can be easily detected by comparing the single features in the merging process. The detection of conflicts in

Table 1. Possible conflicts in the consolidation of knowledge models A and B

Conflict between	**Characteristic A**	**Characteristic B**
Feature Types	Mandatory	Optional
Tree Relationships	XOR	OR
Cross-Tree Relationships	requires supports	excludes hurts

Cross-Tree Relationships is more computation-intensive as it requires the traversal of the whole feature model tree. Nevertheless, this effort is justified as faulty *Cross-Tree Relationships* can lead to impossible business model instances. To resolve the knowledge conflicts, the *Business Developer* can store his preferred decisions into his *Company Business Knowledge Model* because these elements will overwrite the knowledge of the *Domain Experts* at the development of the business model.

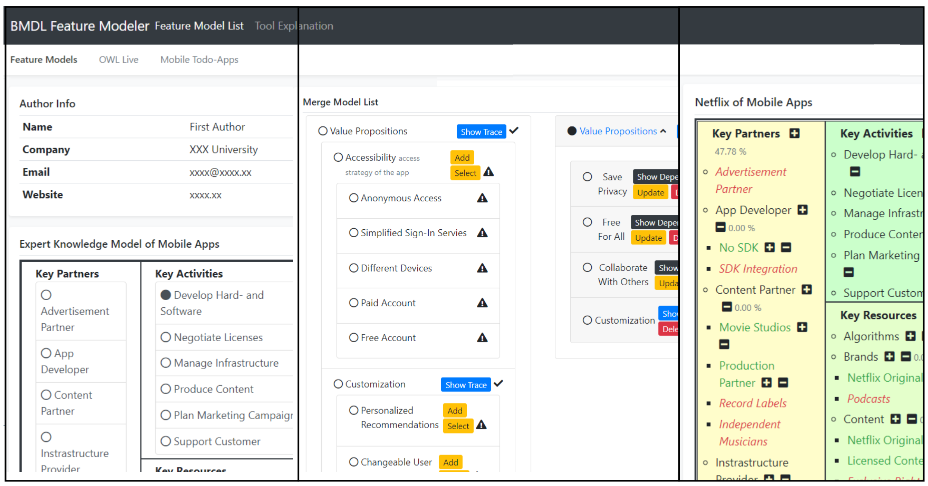
3.3 Integrating Expert Knowledge into BMDTs

After the *Business Developer* has selected the expert knowledge and resolved potential conflicts, the *Consolidated Expert Business Knowledge Model* needs to be integrated into the business model development process. For this, we provide three blueprints how developers of BMDTs could use those expert knowledge in their corresponding tools:

– **Discover Business Elements:** During the design of new business models, expert knowledge can be used as a library to discover possible business model elements that the *Business Developer* can use. By providing descriptions for all elements, the library ensures a common understanding between different *Business Developers*. Moreover, expert knowledge can be used to check the designed business model against the recommendation of experts, which supports the *Business Developer* in building effective business models.
– **Suggesting Business Patterns:** The existing expert knowledge can also be used to suggest possible business model improvements to the *Business Developer*. For this, the tool can suggest possible business model patterns if parts of the patterns are already used in the business model. Moreover, the tool can analyze the strength (modeled as support-relationship) and weaknesses (modeled as hurt-relationship). This can support the *Business Developer* in focusing on the most critical parts of the business model.
– **Comparing Business Models:** Finally, the *Business Developer* can compare their designed business models with examples of expert knowledge. Here, it is possible to directly choose competitors' business models to analyze competitive advantages by differences in the selected elements. Moreover, it is possible to search for similar existing business models in the whole library. These companies, in turn, can be analyzed by the *Business Developer* to gather more insights for his own business.

4 Technical Implementation

In this section, we show the technical implementation of our approach. For this, we create a ready-to-use *Expert Business Domain Knowledge* modeling language[1] and integrate the concept of the knowledge consolidation together with the blueprints in a Business Model Development Tool called *BMDL Feature Modeler*[2].



a) Adding Expert Knowledge b) Merging Knowledge Sources c) Comparing Business Models

Fig. 5. Overview of the BMDT with examples on *(a) Adding Expert Knowledge*, *(b) Merging Knowledge Sources* and *(c) Comparing Business Models*

The *BMDL Feature Modeler*, which is shown in Fig. 5, is based on an existing tool that we already presented in [18]. Here, we introduced the concept of combining the engineering process of Software Product Lines with the structure of the Business Model Canvas to model business models. In this paper, we extend the tool for modeling expert knowledge (see Fig. 5 (a) for creating an expert knowledge model), consolidate the knowledge models, and detect conflicts (see Fig. 5 (b) for detecting knowledge conflicts) together with the blueprint of how the knowledge can be used (see Fig. 5 (c) for a comparing business models). In the following subsection, we give details on the implementations behind these concepts. Moreover, the publish the source code of our tool[3] so that it is usable and extensible by the whole information systems community.

[1] Language Specification: `https://github.com/sebastiangtts/bmdl-feature-modeler/tree/master/specification/`
[2] Online Version: `https://sebastiangtts.github.io/bmdl-feature-modeler/`
[3] Source Code: `https://github.com/sebastiangtts/bmdl-feature-modeler/`

4.1 Modeling of Expert Knowledge

The modeling of the *Business Knowledge Model* is based on the JavaScript Object Notation (JSON). JSON is a lightweight file format that uses simple key-value pairs and arrays. We use JSON as it is a wide-accepted standard for data transmission in web applications. Moreover, the file easy to read and write for humans and easy to parse and generate for software. To support the structuring of those data, we use JSON Schema. JSON Schema[4] provides a vocabulary that allows the annotation and validation of JSON documents. This standardization, in turn, allows us to provide compatibility and data exchange between different BMDTs. The JSON can be created with a graphical editor inside the *BMDL Feature Modeler* (see Fig. 5 (a) for creating an expert model) or any other text editor (see Fig. 6 (b) for a textual document).

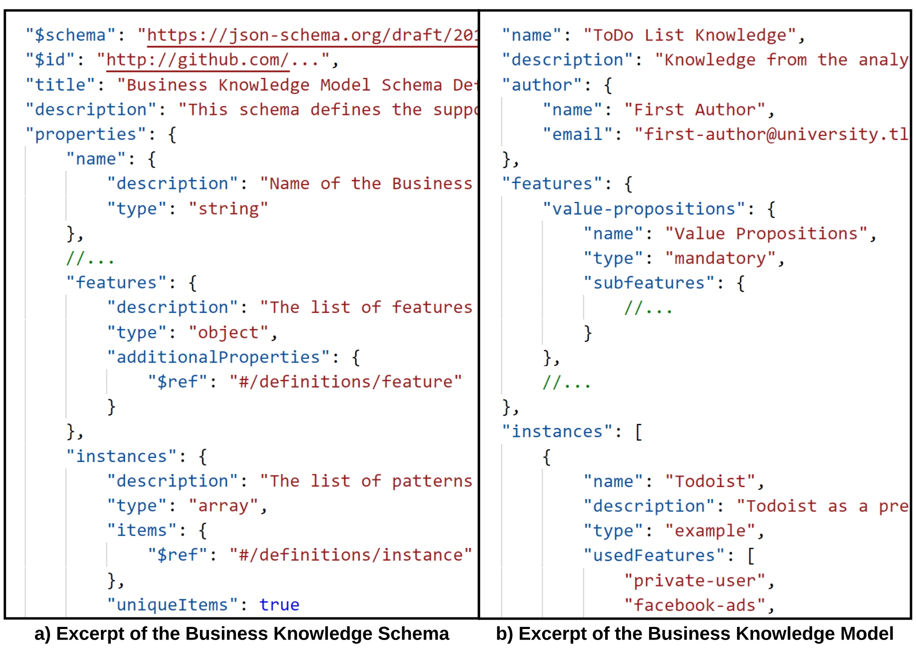


Fig. 6. Excerpts of the *Business Knowledge Schema* and developed *Business Knowledge Model* based on code snippets

A fragment of our schema and a valid model is shown in Fig. 6. While the *Business Knowledge Schema* (see Fig. 6 (a)) provides formalization for valid models that are based on our meta-model in Fig. 4, the *Business Knowledge Model* (see Fig. 6 (b)) shows a possible valid model of an expert. Inside the schema, which is based on the meta-model in Fig. 4, we define a unique identifier

[4] Website of JSON Schema: `https://json-schema.org/`

together with the properties of general model information, the corresponding author, the features of the model, and possible instances. The features, which are nested in each other, have an identifier, a name, properties, and relationships to other features (based on their identifiers). The instances have a name, a type (example or pattern), and a list of feature identifiers the instance is using. While the modeling is possible within our tool, the full schema and exemplary model together with a detailed explanation can be accessed in our repository.

4.2 Knowledge Consolidation and Conflict Detection

After modeling the expert knowledge, we need to consolidate this knowledge with the business knowledge to make it usable within the business model development process. For this, we need to merge the features and relationships of both models (see Fig. 5 (b) for merging the business knowledge and the expert knowledge). Instead of physically merging those features, we create virtual trace links between the models in the *BMDL Feature Modeler*. Virtual trace links are additional links between both knowledge bases. This, in turn, simplifies adding, modifying, and removing the different expert models. An example of using these trace links can be seen in Fig. 7. While in this section, we describe the merging of the *Business Knowledge Model* with a single *Expert Business Knowledge Model*, the steps can be repeated for every other *Expert Business Knowledge Models* to create a homogeneous knowledge base.

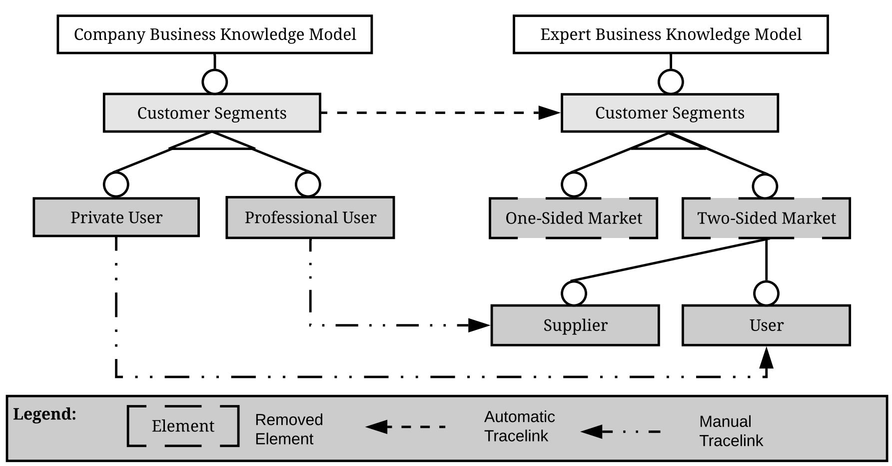


Fig. 7. Example of the Knowledge Consolidation based on Automatic and Manual Trace Links

At the beginning of the step, all nine *Components* of the models are automatically merged because they exist in both models (see *Customer Segments* in Fig. 7). After that, the *Business Developer* manually selects the *Elements* he wants to use. If the *Element* does not exist within the *Company Business Knowledge Model*, it can directly be added to the hierarchy of the model (see removing of *One-Sided Market* in Fig. 7). Otherwise, the attributes (*Type*, *TreeRelationship*) of both *Elements* need to be compared to detect possible conflicts. Moreover, the *Business Developer* can link *Elements* of the *Expert Business Knowledge Model* directly to *Elements* of the *Business Knowledge Model*. This is used to overcome the restrictions of the hierarchy and merge equal *Elements* with different namings (see trace link from *Private User* to *User* in Fig. 7). After all *Elements* have been added to the *Company Business Knowledge Model*, the *Business Developer* also needs to add the *CrossTree-Relationships* between both models. Here, we need to check all *CrossTree-Relationships* where both *Elements* are merged with the *Business Knowledge Model* for potential conflicts (e.g. conflict of hurts- and supports-relationships). To avoid following cycles in the *CrossTree-Relationships*, the whole traversing of the model is needed. The whole step, which is used for a single expert model, is now repeated for all expert models.

4.3 Integrating Expert Knowledge into BMDTs

After consolidating the knowledge of the different experts, we have a single *Business Knowledge Model*, which can be used to support the business model development process. For this, we have conceptualized three different blueprints in the last section.

In *Discover Business Elements*, we want to show the business developer the business elements he can use. For that, we provide at each component and element a button to open a list of subfeatures with a name and explanations. Moreover, we check the business model against the relationships in the *Business Knowledge Model* to show conformance errors between both.

In *Suggesting Business Patterns*, we want to suggest business model improvements to the developer. For that, we compare the elements in the patterns with the elements in the business models to show existing patterns and provide recommendations for patterns where single elements are missing. Moreover, we highlight strengths and weaknesses in the business model according to the hurts- and supports-relationships.

The last blueprint is *Compare Business Models*, where we compare our own business model with other business models based on a heatmap (see Fig. 5 (c) for a comparing business models). For that, we provide Algorithm 1 to calculate the distance between the features sets OF (Own Features) and CF (Comparison Features) based on a feature model FM. In COMPAREMODELS(FM, OF, CF), we sum up the similarities of each component to get the overall similarity of the business models. In COMPAREFEATURE(F, OF, CF), we compare the similarity of a single feature with all its subfeatures. Here, we halved the similarity weight in each hierarchy level because elements in lower levels are less important than the upper ones.

Algorithm 1 Comparison of different business models

```
function CompareModels(FM, OF, CF)                              ▷ Compare Business Models
    similarityCounter, similarityScore ← 0
    for feature in FM.features do
        similarityScore ← CompareFeature(feature, OF, CF)
        similarityCounter ← similarityCounter + similarityScore
        print feature.name+": "+similarityScore                  ▷ Component Similarity
    end for
    print "Business Model Canvas: "+ similarityCounter/9          ▷ Canvas Similarity
end function

function CompareFeature(F, OF, CF)                              ▷ Compare Business Elements
    similarityCounter, similarityScore, featureCounter ← 0
    for subfeature in F.subfeatures do
        if subfeature in OF and subfeature in CF then
            featureCounter ← featureCounter + 1
            similarityScore ← 0.5+0.5× CompareFeature(subfeature, OF, CF)
            similarityCounter ← similarityCounter + similarityScore
            print subfeature.name+": "+similarityScore           ▷ Element Similarity
        else if subfeature in OF or subfeature in CF then
            featureCounter ← featureCounter + 1
        end if
    end for
    return featureCounter > 0 ? similarityCounter/featureCounter : 1
end function
```

5 Application to Local Event Platform

In this section, we show how the approach can be applied to a concrete usage scenario. For this, we first instantiate our approach on top of business models for a local event platform (Sect. 6.1) and second discuss the current limitations of the approach (Sect. 6.2).

5.1 Instantiation

We show the applicability of our approach by providing an instantiation on OWL Live. OWL Live is a local event platform created in the OWL culture portal's research project[5]. This research project aims to establish a local area event platform that the project partners should sustainably operate. The value of the platform is to aggregate event information from different sources based on machine learning algorithms. OWL Live is a two-sided market between event providers and event visitors that both have to be considered during business model development. At the beginning of the instantiation, we interview the responsible project manager to gather information about the platform. According to Teece [34], we ask questions about the market, the possible competitors, and

[5] Project Website: `https://www.sicp.de/en/projekte/owlkultur-plattform`

the own niche. After the interview, we use the information to create different *Expert Business Knowledge Models* and the *Company Business Knowledge Models.* After consolidating that knowledge, we derive three possible *Business Models* for the platform.

We use the *Expert Business Knowledge Models* to store the information about the market and the possible competitors. For the market, we first cover mobile applications in general. Here, we use our existing feature model for business models of mobile applications as introduced in [17]. Because the model allows just standard feature models relationships, we add hurts- and supports-relationships (e.g., In-App Ads hurts Privacy) to the model. Moreover, we add existing patterns (e.g., Low-Price Strategy) and the existing models as examples (e.g., Spotify) to the model. After that, we create additional expert models for application fields related to the platform's concept. We gather our information by analyzing the business model of a subset of existing companies in that field. The analyzed fields were content aggregations (e.g., Rotten Tomatoes), which aggregate content from different sources, social media networks (e.g., Instagram), which provide interactions of a mass amount of users, and trending apps (e.g., Clubhouse), which should provide us information about current usage trends. For the possible competitors, we analyze event apps (e.g., Eventim), which act in a broader range than the platform, and local competitors (e.g., local newspaper), which provide an alternative to the usage of the platform. In total, we created six expert knowledge models.

We use the *Company Business Knowledge Model* to store information about the niche that the platform should have. This information is mostly obtained from the project manager. It contains ideas for specialized customer segments (e.g., culture enthusiasts), new customer relationships (e.g., customer contact over culture offices), new revenue streams (e.g., usage of sponsorships), and enhanced value propositions (e.g., route approximation to event).

After consolidating that knowledge, we use it to derive three different *Business Models.* First, we derive a type of content aggregator, where a mass amount of local events is crawled to gain interest for a mass market of users. Based on that, revenue streams of personalized advertisements and affiliate links to existing ticket sellers could be established. Second, we derive a type of ticker seller, where the focus is mainly on small local events. The customer relationships could be arranged personally, and a commission fee could generate revenue. Third, we derive a type of sponsored platform, where revenue is gained from private and public sponsorships. Based on that, value propositions of privacy-friendly usage and independent prioritization could be established. Using our tool, all developed business models can be directly compared to the event app and the local competitors to analyze a competitive advantage.

5.2 Discussion

With the implementation and its instantiation, we show the applicability of our approach. Nevertheless, while conducting the instantiation, we found some limitations with respect to the *Business Knowledge Generalization*, the *Business Process Modeling* and the *Instantiation Restrictions*.

For the *Business Knowledge Generalization*, we currently based our *Business Knowledge Model* on the Business Model Canvas (BMC). While the BMC is widely used for business model innovation, other canvas structures support other steps of the innovation process (e.g., Value Proposition Canvas for identifying the needs of the customer) or special types of business models (e.g., Platform Canvas for platform business models). Therefore, we want to improve our *Business Knowledge Model* by supporting freely definable canvas structures in the future.

For *Business Process Modeling*, we currently allow the execution of steps of the innovation process (e.g., adding expert knowledge, develop business model) concurrent with each other, which increases the complexity of the approach. Moreover, it provides the business developer less guidance about methods to derive the knowledge of the business knowledge model. Therefore, we want to extend our approach by providing a stepwise creation and validation of business models.

For *Instantiation Restrictions*, we applied our approach to the development of business models of a local event platform. Although this allowed us to demonstrate and evaluate all steps of our approach, it has the limitation that we combined the domain expert and the business developer in one person. This results in less knowledge to consolidate and conflicts to detect. Therefore, we want to conduct workshops where business developers must use existing expert knowledge to validate our approach further.

6 Related Work

In this section, we show the related work of our approach. We divide this work into *Knowledge Modeling* and *Business Model Development Tools*.

In the area of *Knowledge Modeling*, current languages for business modeling do not support the meta-modeling of business model knowledge [20]. Therefore, we look into the similar topic of requirements engineering which also provides the foundation for feature models. In goal-oriented requirements engineering [35], the different user needs are modeled as goals with relationships between them. Here, languages like iStar [9] or KAOS [35] provide different semantic relationships like decompositions and contributions types (e.g. help, hurt) between the goals to structure them. Because these requirements can come from many sources, tools for requirements consolidation have already been developed [25]. Moreover, this consolidation is also used in Software Product Lines with the merging of feature models [2]. Nevertheless, these approaches are built for requirements engineering and cannot directly be transferred to the different contexts of business modeling (e.g. modeling business pattern). Moreover, they are not used to reuse gained expert knowledge.

In the area of *Business Model Development Tools*, current tools in practice do not support the usage of expert knowledge [33]. Therefore, we look into current research which mostly develops design principles for future BMDTs. The Business Model Assistance System [10] uses a reference database of existing business models for comparison with the own business model. The Business Model Developer [5] is a domain-specific approach with a shared vocabulary based on a taxonomy and uses semantic relationships between the elements for financial calculations. The concept of semantic relationship is also used by Business Model Analyzer [4] to support the business model comprehension. The Green Business Model Editor [32] uses existing schemas to provide patterns for sustainable business models. The idea of the sustainable business pattern, which is modeled through a taxonomy, is also implemented by the Smart Business Modeler [22]. The Computer-Aided Business Model Design [30] introduces a concept for bringing different business developer experience levels into account. Here, novices are supported in coherent modeling, experts model the interactions of business model elements, and masters aim to evaluate different business model alternatives. Nevertheless, these approaches are made for knowledge models that are made by a single expert and do not support multiple knowledge sources and a corresponding knowledge consolidation.

7 Conclusion and Outlook

Business model innovation is a creative task that often requires the external knowledge of experts. While this expert knowledge is easily accessible in workshops, current BMDTs do not support reusing this knowledge. This expert knowledge, in turn, could improve the quality of the developed business models. In this paper, we present an approach to consolidate the knowledge of different experts to support the business model innovation process. With our approach, different domain experts can model their expert knowledge based on a ready-to-use modeling language. Business developers, in turn, can model the company knowledge and consolidate that knowledge with expert knowledge. This consolidated knowledge can then be used in various ways during the business model development. For this, we develop different blueprints to extend existing business model development tools. We implement the whole approach in an open-source tool and show the applicability with an exemplary instantiation for a local event platform.

Our future work is threefold and deals with improving the current limitations in the discussion of our instantiation. First, we want to improve the current limitations in terms of business knowledge generalization by providing support for different canvas structures. This will ensure a broader usage of the modeling language and tooling. Second, we want to improve business process modeling by providing stepwise execution methods for developing and validating business models. This will provide business developers additional support in the business model development. Third, we want to work on the instantiation restrictions by conducting workshops with business developers to derive their own business models. This will increase the validity of our approach in real-life settings.

References

1. Alberts, B.T., Meertens, L.O., Iacob, M.E., Nieuwenhuis, L.J.M.: A Meta-Model Perspective on Business Models. In: Business Modeling and Software Design, vol. 142, pp. 64–81. Springer (2013)
2. Alves, V., Schwanninger, C., Barbosa, L., Rashid, A., Sawyer, P., Rayson, P., Pohl, C., Rummler, A.: An Exploratory Study of Information Retrieval Techniques in Domain Analysis. In: 12th International Software Product Product Line Conference (SPLC), pp. 67–76. IEEE (2008)
3. Apel, S., Batory, D., Kästner, C., Saake, G.: Feature-Oriented Software Product Lines. Springer, Heidelberg (2013)
4. Augenstein, D., Fleig, C., Maedche, A.: Development of a Data-Driven Business Model Transformation Tool. In: Designing for a Digital and Globalized World, vol. 10844, pp. 205–217. Springer (2018)
5. 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 (2017)
6. Chesbrough, H.: Business model innovation: it's not just about technology anymore. Strategy & Leadership **35**(6), 12–17 (2007)
7. Clements, P., Northrop, L.: Software product lines: Practices and patterns. Addison-Wesley, Boston, 7th edn. (2009)
8. Cosenz, F., Noto, G.: A dynamic business modelling approach to design and experiment new business venture strategies. Long Range Planning **51**(1), 127–140 (2018)
9. Dalpiaz, F., Franch, X., Horkoff, J.: istar 2.0 language guide (2016)
10. Di Valentin, C., Emrich, A., Werth, D., Loos, P.: Business Modeling in the Software Industry: Conceptual Design of an Assistance System. In: Practice-Driven Research on Enterprise Transformation, vol. 151, pp. 34–45. Springer (2013)
11. Ebel, P., Bretschneider, U., Leimeister, J.M.: Leveraging virtual business model innovation: a framework for designing business model development tools. Information Systems Journal **26**(5), 519–550 (2016)
12. Fritscher, B., Pigneur, Y.: Supporting Business Model Modelling: A Compromise between Creativity and Constraints. In: Task Models and Diagrams for User Interface Design, vol. 5963, pp. 28–43. Springer (2010)
13. Gassmann, O., Frankenberger, K., Csik, M.: The business model navigator: 55 models that will revolutionise your business. Pearson, Harlow (2014)
14. Geissdoerfer, M., Bocken, N.M., Hultink, E.J.: Design thinking to enhance the sustainable business modelling process – A workshop based on a value mapping process. Journal of Cleaner Production **135**, 1218–1232 (2016)
15. General Electric Inc: GE Global Innovation Barometer 2018, `https://www.ge.com/reports/innovation-barometer-2018/`
16. Gordijn, J., Akkermans, H.: Designing and evaluating e-business models. IEEE Intelligent Systems **16**(4), 11–17 (2001)
17. Gottschalk, S., Rittmeier, F., Engels, G.: Intertwined Development of Business Model and Product Functions for Mobile Applications: A Twin Peak Feature Modeling Approach. In: Software Business, pp. 192–207. Springer (2019)
18. Gottschalk, S., Rittmeier, F., Engels, G.: Hypothesis-driven Adaptation of Business Models based on Product Line Engineering. In: International Conference on Business Informatics (CBI). IEEE (2020)

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. International Journal of Operations & Production Management **36**(10), 1382–1406 (2016)
20. John, T., Kundisch, D., Szopinski, D.: Visual Languages for Modeling Business Models: A Critical Review and Future Research Directions. In: Proceedings of the 38th International Conference on Information Systems (ICIS). AIS (2017)
21. Kästner, C., Apel, S., Kuhlemann, M.: Granularity in software product lines. In: 13th International Conference on Software Engineering (ICSE). p. 311. ACM (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: Sustainable Business Models, pp. 25–60. Springer (2019)
23. Lüdeke-Freund, F., Carroux, S., Joyce, A., Massa, L., Breuer, H.: The sustainable business model pattern taxonomy—45 patterns to support sustainability-oriented business model innovation. Sustainable Production and Consumption **15**, 145–162 (2018)
24. Meertens, L.O., Iacob, M.E., Nieuwenhuis, L., van Sinderen, M.J., Jonkers, H., Quartel, D.: Mapping the Business Model Canvas to ArchiMate. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC). ACM (2012)
25. Nagappan, M., Shihab, E.: Future Trends in Software Engineering Research for Mobile Apps. In: 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). pp. 21–32. IEEE (2016)
26. Osterwalder, A.: The Business Model Ontology: A Proposition in a Design Science Approach. Dissertation, University of Lausanne, Lausanne (2004)
27. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken (2010)
28. Osterwalder, A., Pigneur, Y.: Designing Business Models and Similar Strategic Objects: The Contribution of IS. Journal of the Association for Information Systems **14**(5), 237–244 (2013)
29. Osterwalder, A., Pigneur, Y., Tucci, C.L.: Clarifying Business Models: Origins, Present, and Future of the Concept. Communications of the Association for Information Systems **16** (2005)
30. Pigneur, Y., Fritscher, B.: Extending the Business Model CanvasA Dynamic Perspective. In: Proceedings of the Fifth International Symposium on Business Modeling and Software Design. pp. 86–95. SCITEPRESS (2015)
31. Reman, G., Hanelt, A., Tesch, J., Kolbe, L.: The Business Model Database - A Tool for Systematic Business Model Innovation. International Journal of Innovation Management **21**(01) (2017)
32. Schoormann, T., Behrens, D., Knackstedt, R.: Design Principles for Leveraging Sustainability in Business Modelling Tools. In: Twenty-Sixth European Conference on Information System. AIS (2018)
33. Szopinski, D., Schoormann, T., John, T., Knackstedt, R., Kundisch, D.: Software tools for business model innovation: current state and future challenges. Electronic Markets **60**(11), 2794 (2019)
34. Teece, D.J.: Business Models, Business Strategy and Innovation. Long Range Planning **43**(2-3), 172–194 (2010)
35. van Lamsweerde, A.: Goal-oriented requirements engineering: a guided tour. In: International Symposium on Requirements Engineering. pp. 249–262. IEEE (2001)