Business Models of Store-Oriented Software Ecosystems: A Variability Modeling Approach*

Sebastian Gottschalk, Florian Rittmeier, and Gregor Engels

Paderborn University, Paderborn, Germany
{sebastian.gottschalk,florianr,engels}@uni-paderborn.de

Abstract. In the last years, store-oriented software ecosystems are gaining more and more attention from a business perspective. In these ecosystems, third-party developers upload extensions to a store which can be downloaded by end users. While the functional scope of such ecosystems is relatively similar, the underlying business models differ greatly in and between their different product domains (e.g. Mobile Phone, Smart TV). This variability, in turn, makes it challenging for store providers to find a business model that fits their own needs.
To handle this variability, we introduce the Business Variability Model (BVM) for modeling business model decisions. The basis of these decisions is the analysis of 60 store-oriented software ecosystems in eight different product domains. We map their business model decisions to the Business Model Canvas, condense them to a variability model and discuss particular variants and their dependencies. Our work provides store providers a new approach for modeling business model decisions together with insights of existing business models. This, in turn, supports them in creating new and improving existing business models.

Keywords: Software Ecosystems · Business Models · Variabilities

1 Introduction

In the last years, software ecosystems are gaining more and more attention from a business perspective. What started with Apple's iOS and Google's Android in the area of mobile phones [5] is transferred to an increasing number of product domains. In the literature, many definitions of software ecosystems exist [14]. For the purpose of this paper, we use the definition of Bosch et al. who define software ecosystems as "a software platform, a set of internal and external developers and a community of domain experts in service to a community of users that compose relevant solution elements to satisfy their needs" [1]. Most of these software ecosystems are store-oriented software ecosystems [10].

In store-oriented software ecosystems (see Figure 1), the store provider, third-party developers, and end users are interacting with each other through a store interface. The store provider provides a software platform in form of a store

* This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing"(CRC 901).

Preprint, cite this paper as:
Gottschalk S., Rittmeier F., Engels G. (2019) Business Models of Store-Oriented Software Ecosystems: A Variability Modeling Approach. In: Shishkov B. (eds) Business Modeling and Software Design. BMSD 2019. Lecture Notes in Business Information Processing, vol 356. Springer, Cham

The final authentificated version is available online at:
https://doi.org/10.1007/978-3-030-24854-3_10

with different features. Examples of these features are a catalog of extensions together with the possibility to rate and review them. Moreover, he can develop extensions to publish them in the store. This development and the publication is also possible for third-party developers who try to reach end users with their extensions. The end users can use the functions of the store and execute extensions of the store provider and third-party developers.

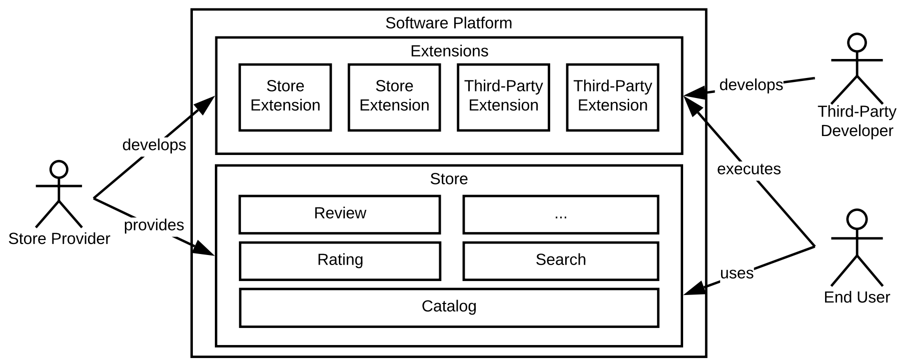

Fig. 1. Overview of a Store-Oriented Software Ecosystem

While the functional scope of such ecosystems is relatively similar and the technological decisions are well-understood [10], less research focused on business models of such ecosystems. For the term of the business model, we use the definition of Osterwalder et al. who point out that "a business model describes the rationale of how an organization creates, delivers, and captures value" [19]. Various software ecosystems have different business models, which are used more and more to differentiate in and between product domains [3]. In the area of software ecosystems, the trend of using a store as interface is strengthened by the increasing connectivity of products [4] and the rising amount of platform business models [20]. Examples of these business model decisions are the hardware-bundling of Apple's iOS, the platform-independence of Valve's gaming store STEAM[1] or the subscription model of Sony's PlayStation. A good example of the impact of wrong business model decisions is Nokia's Symbian OS, which failed with business model decisions like high commission fees for third-party developers, incompatibilities between different versions of the ecosystem and poor user experience for end users [2]. This, in turn, leads us to the question of the different business model decisions that exist for the individual product areas and what variabilities and dependencies can be derived from them.

To answer this question, we analyze 60 stores in eight different product domains and identify the most important business model decisions using the taxonomy development method of Nickerson et al. [18]. To structure these decisions, we introduce the Business Variability Model (BVM) based on the Business

[1] https://store.steampowered.com/

Model Canvas (BMC) and its nine building blocks (Customer Segments, Value Proposition, Channels, Customer Relationships, Key Activities, Key Resources, Key Partners, Revenue Streams, Cost Structure). The advantage of our Business Variability Model compared to the Business Model Canvas is the possibility to differentiate between mandatory and optional business model decisions and to model dependencies between these decisions. The result of our work is a variability model for these business model decisions as well as a deeper analysis of the variants for revenue streams and their dependencies to the channels. Our work supports store providers in developing new and improving existing business models for store-oriented software ecosystems.

In the following, Section 2 considers the related work of the topic. Section 3 describes our used research method and Section 4 shows the derived variability model. Based on the variability model, the variants of revenue streams together with their dependencies to the channels are analyzed in a more fine-graded way. In Section 5 we show the validity of our variability model by describing the ecosystem of Sony's PlayStation. After that, we discuss our results in Section 6. Finally, a conclusion is given in Section 7.

2 Related Work

There are already several articles in the literature that deal with individual parts of business models of store-oriented software ecosystems. For example, there are articles on the general strategy of a software ecosystem [5, 7, 8, 17, 22], whose individual customer segments are analyzed [5, 8, 10, 13, 15, 17, 22] and the associated value propositions [5, 6, 8, 13, 15, 17, 21, 22] are discovered. To reach customers, they need to be acquired [5, 6, 13, 17] and relationships need to be maintained [5,13,17,21,22]. Income [5,6,10,15,17,21,22] is generated by providing services to customers. In order to operate the ecosystem, partnerships need to be formed [5, 6, 10, 17], activities must be carried out [5, 15, 17] and resources must be created [5, 6, 8, 10, 15, 17]. This leads to costs [5, 6, 17] for the store provider. However, while many contributions are limited to individual business areas or the direct comparison of different ecosystems in case studies, only a few contributions [5,8,10,17] attempt to identify the key variabilities in the business models.

Goncalves et al. [5] describe the ongoing platformization process in the mobile network area and how mobile network operators can benefit from that. To do this they analyze different kinds of software ecosystems and group them into four patterns by the variabilities in customers and assets control. These patterns are Enabler Platform, System Integrator Platform, Neutral Platform and Broker Platform. For each pattern they give a small overview of their features, discuss success factors and point out how these patterns can be adapted to mobile operators.

Mueller et al. [17] analyze the competition among different mobile app stores. To do that they are using a literature review to point out the most important business parts of app stores like store features, value propositions, revenue and costs streams together with the main stakeholders of the store. They model the value streams around the customers and partners of the ecosystems and compare the value influence of single partners to the value of the related mobile app store.

Jansen et al. [8] analyze the characteristics, policies, and features of an app store for the store owner, end users and third-party developers based on different case studies. After introducing the core features of app stores, they point out user-focused features (App Findability, App Quality, App Store Usability) and developer-focused features (Feedback Potential, Monetization Potential, App Store Usability, Visibility) as variation points with different variants.

Jazayeri et al. [10] propose a variability model for architectural design decisions of store-oriented software ecosystems. They combine a systematic literature review with the examination of software ecosystems to conduct variabilities in the areas of business, application and infrastructure decisions. By focussing on the technical aspects of these ecosystems, they give, for the business perspective, just an abstract view of the variabilities of complementary partnerships, fees, openness factors, and license agreements. After applying this variability model to a set of store-oriented software ecosystems [11], they derive three patterns for software ecosystems, namely Resale Software Ecosystems, Partner-based Software Ecosystems, and Open Source Software-Based Ecosystems.

3 Research Method

In the paper, we develop a variability model for business model decisions. Therefore, we combine the concept of variability modeling with the structure of the Business Model Canvas. For the underlying decisions, we are using a taxonomy development method which was proposed by Nickerson et al. [18]. The method can be used to classify objects based on their common characteristics. To use the method, we need to define meta-characteristics and ending-conditions together with empirical-to-conceptional and conceptional-to-empirical iteration steps.

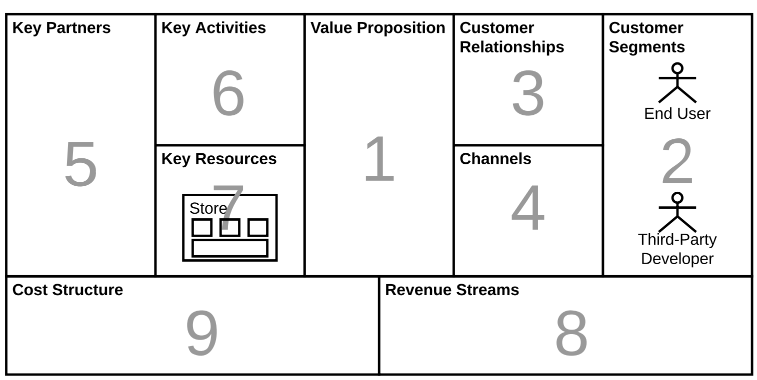

Fig. 2. Business Model of the Store Provider

The meta-characteristics are the most comprehensive characteristics that can be used as the basis for the choices in the taxonomy. Based on this meta-characteristics we are running combinations of empirical-to-conceptional and conceptual-to-empirical iterations. After each iteration, the taxonomy is checked against objective and subjective ending conditions. While objective ending conditions can be clearly assessed, subjective ending conditions leave space for interpretation. At the time all ending conditions are fulfilled, we are using the taxonomy as the structure for our variability model. After deriving this variability model, we create the dependencies between the individual variation points and variants (see Figure 3).

To start with the taxonomy development, we need at first to define the meta-characteristic. As we are using the Business Model Canvas as our structure to define the business decisions, we are using the nine building blocks (see Figure 2) as start points for our meta-characteristics. To focus on the store as the key element and its dependencies to third-party developers and end users, we design the business model of the store provider as a two-sided market with the store as a key resource. Other stakeholders within the business model are defined as key partners of the store provider. This view of a two-sided market is also used often in the literature of software ecosystems [1, 8]. Moreover, we distinguish all variabilities between store provider, third-party developer and end user to separate them by their stakeholder. As an objective condition, we want to examine all selected objectives (Stores, Papers) and as subjective condition, we want to create an appropriate and cross-domain usable model that can be easily extended.

In the empirical-to-conceptional iteration, we analyze a set of 60 store-oriented software ecosystems in eight different product domains with respect to their business model. As product domains, we choose a broad area of domains where the concept of store-oriented software ecosystems is already successfully implemented through different business models. The following list provides an overview of the different product domains and their included software ecosystems:

- **Mobile Phone**: Amazon Appstore, Apple App Store, Aptiode, F-Droid, GetJar, Google Play, Samsung Galaxy Store, SlideMe, UpToDown
- **Video Game Console**: Microsoft Store, Nintendo eShop, Ouya Games, PlayStation Store
- **Smart TV**: Amazon Fire OS Store, Google Play, LG AppStore, Panasonic MyHomeScreen, Roku ChannelStore, Samsung Galaxy Store, Tizen Store, VEWD AppStore
- **Personal Computer**: Canonical Snapcraft, Chocolatey, FlatHub, GNOME Shell Extensions, Mac AppStore, Microsoft Store, Npackd, Plasma Discover, Softtonic
- **Gaming Platform**: EA Origin, EPIC Store, Good Old Gaming, Green Man Gaming, Humble Store, Ubisoft Uplay, Valves STEAM
- **Software Extension**: Chrome WebStore, Eclipse Marketplace, Firefox Addons, Kodi Media Center, Libre Office, Media Portal, Microsoft AppSource, Microsoft Store, Opera Addons, Safari Extension Library, Thunderbird Addons, Visual Studio Marketplace, VLC Media Player

- **Digital Personal Assistent**: Amazon Alexa, Apple's Siri, Google Assistent, Microsoft Cortana, Samsung's Bixby
- **Task Automatization Platform**: Automate.io, IFTTT, Microsoft Flow, Slack, Zapier

For each store, we inspect the website as an end user and if possible as a third-party developer to derive the different business model decisions regarding to the BMC. Furthermore, we analyze news articles together with the technical documentations of the ecosystems.

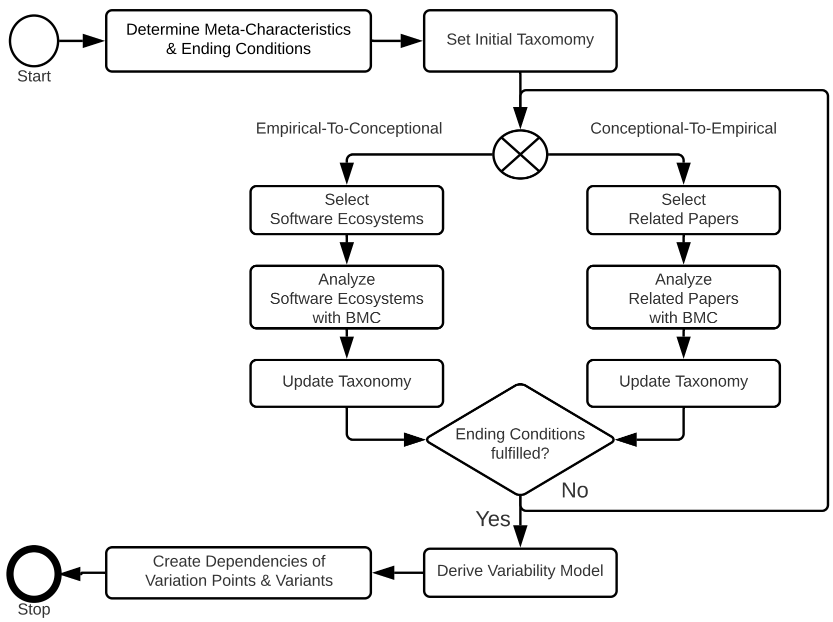

Fig. 3. Development Process of the Variability Model (based on Nickerson et al. [18])

In the conceptual-to-empirical iteration, we analyze related work which based on the systematic literature review of Jazayeri et al. [9] and additional papers provided by our domain knowledge. In the literature review, they analyze the different features of IT Service Markets. One of the extracted features is the business model whose selected papers we are using as our starting papers. We inspect each paper on their business model decisions regarding the BMC and discuss the results in terms of their indirect influence on other parts of the business model decisions.

4 Business Variability Model

In this section, we give an overview of the Business Variability Model (BVM). After we introduce some initial considerations of the variability modeling, we present the variability model for store-oriented software ecosystems. In the following steps, we focus on the revenue streams as a variation group of the business models and discusses the dependencies inside the model regarding the revenue streams and channels.

4.1 Initial Considerations

The modeling of variability in product lines has a long history in the area of software development [23]. A software product line (SPL) comprises several versions of a software system which are based on a shared platform. In some parts of this platform, variabilities are defined, which can be fulfilled with different variants. The approach enables a fast and cost-effective adaptation of software systems. While variability modeling also has a long history in the area of business processes [12], to the best of our knowledge, there was no explicit modeling of variabilities for business model decisions in research.

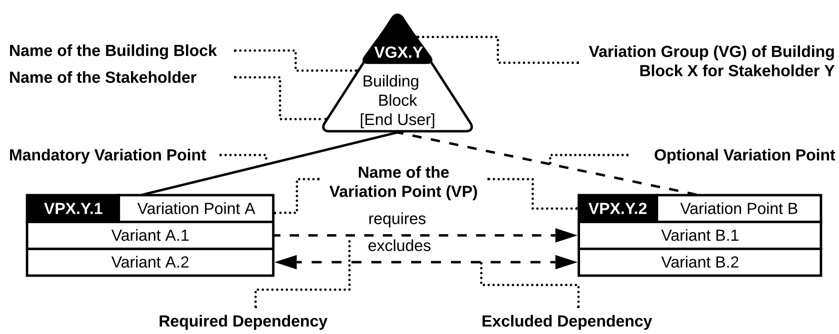

Fig. 4. Legend for Business Variability Model (BVM)

To use the variability modeling in case of business model decisions (see Figure 4), we create the Business Variability Model (BVM) as a modified version of the Orthogonal Variability Model (OVM) by Metzger et al. [16]. With respect to the work of Zhang et al. [24] we are using a three-level variability model with levels called Variation Group (VG), Variation Point (VP) and Variant.

Overall, the model structure can be divided into the nine building blocks of the Business Model Canvas, whereby there can be variabilities for each stakeholder. The structure was simplified to represent a building block and a stakeholder in a single variation group. The first number X of the variation group represents a building block (1 = Value Proposition, 2 = Customer Segments,... ;

see Figure 2 for specific blocks) and the second number Y represents the stakeholder identifier (1 = Store Provider, 2 = End User, 3 = Third-Party Developer). For example, VP1.2 refers to the value proposition of the end user. For each variation group, required and optional variation points are defined. Moreover, the variation points can have required or excluding dependencies to each other. These variation points can be described by different variants, which can have required or excluding dependencies themselves. If a variation point is used by multiple stakeholders, the structure can be simplified by using a x instead of the stakeholder identifier. In this case, the stakeholder can have different variants of the shared variation point. Moreover, we add two grouping strategies (see Figure 5) of grouping variation points and variants to simplify the model structure. By grouping variation points and variants of the same type (mandatory, optional) to each other, the structure can be represented in a compressed way.

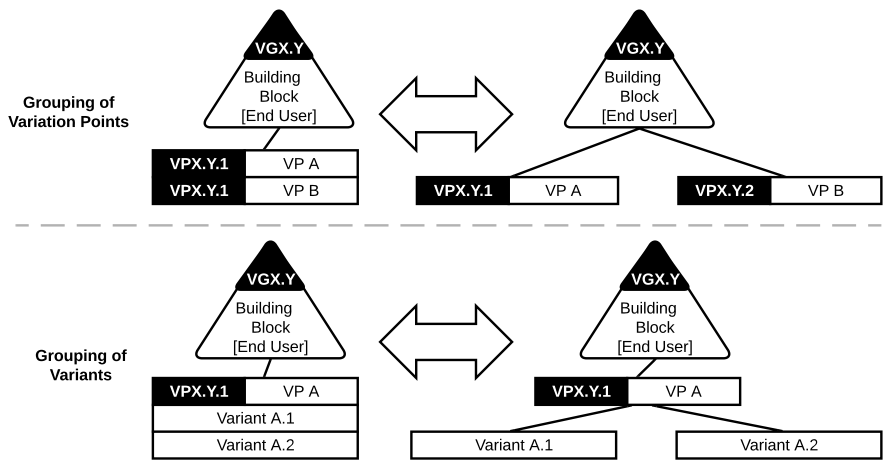

Fig. 5. Grouping of Variation Points and Variants

With the help of this BVM, we are able to create a meta-structure for possible Business Models. Moreover, with the created dependencies we are able to analyze if changes in some parts of the business model lead to changes in or have conflicts with other parts of the business model. Because within this paper we are not able to describe all variation points, variants, and their dependencies, we are focussing on the revenue streams and their dependencies to the channels.

4.2 Modeling of Variation Groups and Variation Points

Within this section, the variability model (see Figure 6) is presented. In the model for each business model decision, the most important points are picked out and described through different variation points.

In order to improve the structure of the model, we divide it into the perspectives of Product / Service, Customer, Activity, and Financial.

Within the **Product / Service Perspective**, the *Value Propositions* are described for each customer group that is determined by the store provider. For example, the end user can be promised a catalog with mass-market or high-quality extensions (see V1.2.3). A case of existing ecosystems are the quality guidelines of Apple's iOS which are more restrictive than for Google's Android. In contrast, the third-party developer can be offered a large or specialized group of end users (see V1.3.3). For the specialized group a good example are video game consoles with gamers as the target group.

In the **Customer Perspective**, the different customer segments, the sales channels and the relationships to the customer groups are described. Within the *Customer Segments*, the different customer group characterizations are carried out. For example, the store provider can choose a high-price solution for a small target group or reach a larger target group with a low entry price (see V2.2.1). Within the *Customer Relationships*, the store provider needs strategies to reach the respective customer groups. For example, a change of an ecosystem can be made more difficult for end users by so-called lock-in strategy [13] (see V3.2.2). Finally, it must be decided within the *Channels* how the ecosystem will be delivered to the respective customer. For example, it is possible to bundle an ecosystem with hardware or a software product (see V4.2.1). This bundling can be seen on Sony's PlayStation, where hardware and software are bundled together. In contrast, most web browsers bundle their stores with their software products.

Within the **Activity Perspective**, the partnerships concluded, the activities carried out and the used resources are described. Within the *Key Partners*, partners are included that are necessary for the store provider to achieve its value propositions. For example, it may be necessary to have a partner to deliver the hardware or distribute the software [5] (see V5.1.1). Most software ecosystems are using external partners for the payment process and the cloud infrastructure. The *Key Activities* are used to describe the most important tasks of the store provider. For example, it is necessary for the provider to carry out quality control of the submitted extensions (see V6.1.3). For all stakeholders, there may also be *Key Resources* that are crucial for the use of the ecosystem. For example, third-party developers need developer tools to create their extensions [10] (see V7.3.2).

The **Financial Perspective** is about comparing the revenues and costs for the individual stakeholders. Within the *Revenue Streams*, the stakeholders, which are responsible for the respective flows, are modeled. For example, an end user can purchase a hardware bundle or an extension (see V8.2.1). The *Cost Structure* breaks down which stakeholders generate the respective costs. For example, parts of the revenue can be passed to the third-party developer as commission when an extension is sold (see V9.3.2).

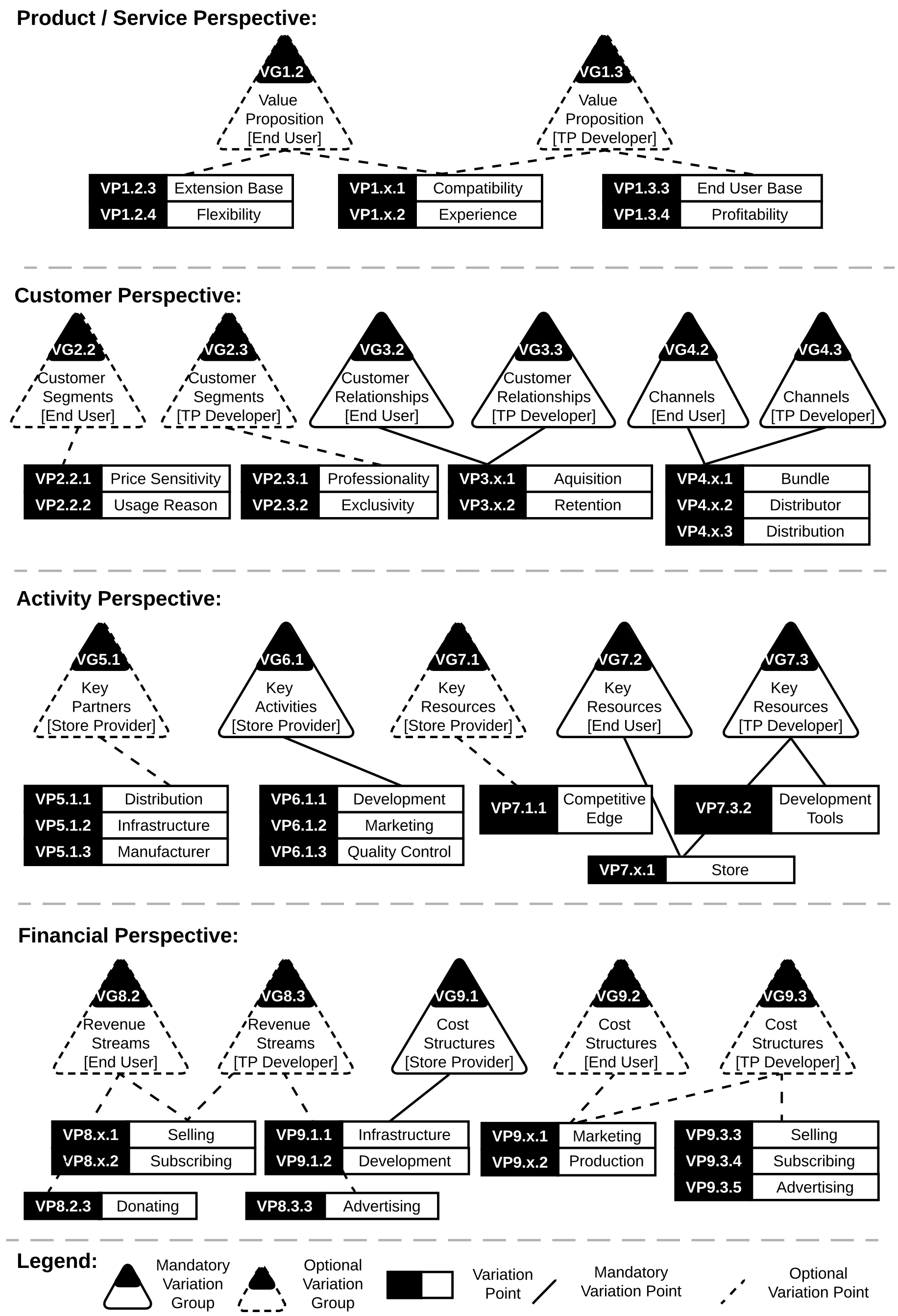

Fig. 6. Variability Model of the Business Model Decisions

4.3 Modeling of Variants

In this section, we focus on the different revenue streams, as on the one hand, they decide on the financial success of store-oriented software ecosystems and on the other hand they show the variety of revenue streams of different product domains.

The variation groups of the revenue streams (see Figure 7) can be divided into the variation points of selling, subscribing, donating and advertising with respect to the end user and the third-party developer.

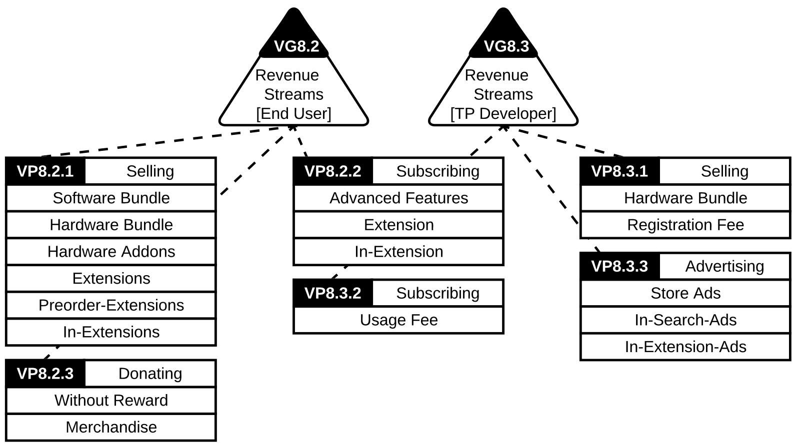

Fig. 7. Variants of Revenue Streams

The store provider can generate revenue from the **End User** by *Selling* different parts of the software ecosystem. If he has bundled the ecosystem, he can sell the software product itself (e.g. Microsoft AppSource with Microsoft Word) or a complete hardware bundle (e.g. Apple Store with iPhone). Moreover, he can sell hardware addons to the existing ecosystem (e.g. Apple Watch). Inside the store, he is able to sell extensions, providing preordering of extensions (e.g Xbox game preordering) or additional in-extensions. To generate a continuous income stream, he can use *Subscribing* offers. Here, the end user can subscribe to advanced features (e.g. PlayStation Plus) together with extensions and in-extensions. The last part of the revenue streams of the store provider are the *Donating* offers, which are often used by non-profit ecosystem provider. Here, the end user can donate without a reward (e.g. Kodi MediaCenter) or get merchandise articles for their support (e.g. Firefox Add-ons).

The store provider can generate revenue from the **Third-Party Developer** by *Selling* different parts of the software ecosystem. If the development of extensions needs separate hardware he can sell it to the developer (e.g. Oculus Rift). Moreover, he can charge a fee for registering as a developer (e.g. Android Developer). This fee can also be paid by *Subscribing* to the ecosystem on an annual basis (e.g. Apple Developer). Moreover, revenue can be generated by *Advertising.* Here, the developer can pay for Store Ads, In-Search-Ads or In-Extensions-Ads.

4.4 Modeling of Dependencies

In this section, we focus on the dependencies between revenue streams and channels. This is because, for channels, in particular, there are many different bundling approaches which have a direct influence on the respective revenue.

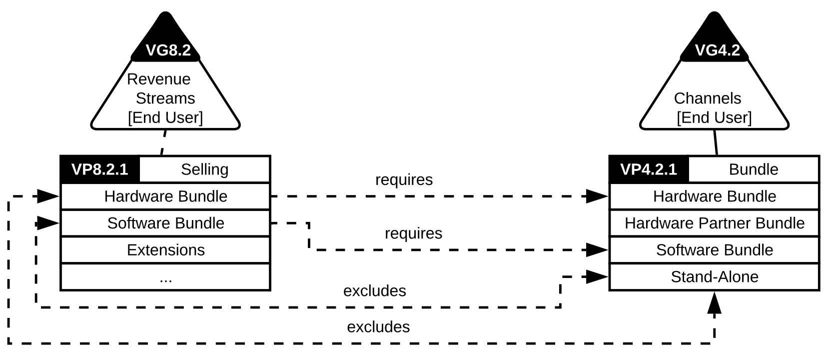

Fig. 8. Dependencies between Revenue Streams and Channels

In the bundling variation point, the store provider is able to choose different bundle options for the **End User** (see Figure 8). The first option is to bundle the specific ecosystems with an own *Hardware Bundle* (e.g. AppStore for Apple's iPhone). If the hardware bundle is not provided by the store provider itself we call it *Hardware Partner Bundle* (e.g. different hardware manufacturers for Google Android). Moreover, the ecosystem can be provided as a *Software Bundle* (e.g. Microsoft AppSource with Microsoft Word) or as *Stand-Alone* (e.g. Valve's STEAM). If the store provider wants to generate revenue from selling a hardware or software bundle, he needs to use the corresponding option in the bundling of the channel. Conversely, he can not generate revenue from hardware and software bundles when he distributes his ecosystem as Stand-Alone.

5 Describing an Existing Ecosystem

To show the validity of our approach, we provide a concrete instance of Sony's PlayStation Ecosystem using our variability model. In Table 1 and 2 we point out the Value Propositions (Va), Customer Segments (CS), Customer Relationships (CR), Channels (Ch), Key Partners (KP), Key Resources (KR), Key Activities (KA), Revenue Streams (RS) and Cost Structure (Co) of the ecosystem. To simplify the tables, we remove all variation points, which are not used in the ecosystem.

Table 1. Describing the PlayStation Ecosystem: Part 1

VG	VP	Store Provider	End User	TP-Developer
Va	**Extension Base**	-	High Amount of Games, Specialized for Gaming	-
Va	**Compability**	-	High Compability though less Fragmentation	
Va	**Experience**	-	Simple User Interface, High Customization	Easy Development Tools
Va	**End User Base**	-	-	High Amount of End Users, Specialized Target Group
Va	**Profitability**	-	-	High Price Points for Games, Sales Events
CS	**Price Sensitivity**	-	High Price Points for Games	-
CS	**Usage Reason**	-	Gaming, Streaming	-
CS	**Professionality**	-	-	Mostly Professional Developer Studios
CS	**Exclusivity**	-	-	Partly Exclusive Developer Studios
CR	**Aquisition**	-	High Amount of Games, Exclusive Games, Hardware Subsidies	High End User Base, Exclusive Deals, Low Entry Fee
CR	**Retention**	-	Lock-In, Exclusive Games, Subscription Model, Gamification	Lock-In, Fair Commission Model
Ch	**Bundle**	-	Hardware-Bundle	-
Ch	**Distributor**	-	Partner-Based-Distribution	Self-Distribution
Ch	**Distribution**	-	Online Shops, Retail Stores, Own Website	Own Website
KP	**Distributor**	Online Shops, Retail Stores	-	-
KP	**Infrastructure**	Amazon Web Services, OpenStack	-	-
KP	**Manufacturer**	Foxconn Tech.	-	-
KA	**Development**	Hardware, Software, SDKs	-	-
KA	**Marketing**	Ecosystem, Playstation Plus	-	-
KA	**Quality Checks**	High Quality	-	-

Table 2. Describing the PlayStation Ecosystem: Part 2

VG	VP	Store Provider	End User	TP-Developer
KR	**Competitive Edge**	Exclusive Brands	-	-
KR	**Store**	-	Store for Downloading Games	Store for Uploading Games
KR	**Development Tools**	-	-	Hardware-Kit, SDK
RS	**Selling**	-	Hardware Bundle, Hardware Addons, Games, Game Addons, Preorder Games	Developer Kit, Developer License
RS	**Subscribing**	Advanced Features (Playstation Plus)	-	-
Co	**Infrastructure**	Cost for Infrastructure	-	-
Co	**Development**	Software, Hardware, Games	-	-
Co	**Marketing**	-	Acquire End User	Acquire Developer
Co	**Production**	-	Hardware	Hardware
Co	**Selling**	-	-	Games, Game Addons, Preorder Games
Co	**Subscribing**	-	-	Games (Playstation Plus)

6 Discussion

In this paper, we create a variability model to model the business model decisions of store-oriented software ecosystems. Because the field of business models is highly dynamic and the valuation of the importance of different business model decisions directly relates to the background experiences of the researchers, the variability model cannot be seen as a closed result for business modeling. Analysis of other researchers or business model decisions of future ecosystems may require changes to the model. Nevertheless, we are convinced that with the analysis of 60 ecosystems and the usage of the taxonomy development method of Nickerson et al. [18], we developed a good starting point for store providers to create new and to improve existing business models.

With the Business Variability Model (BVM), we provide a model to structure these business model decisions. By using the well-accepted Business Model Canvas (BMC) as a starting point, we create a structure which is generic enough to model additional variabilities of other researchers and adopt changes in the future. Therefore, this extendibility is chosen as a subjective ending condition of the taxonomy development.

7 Conclusion and Future Work

In the last years, software ecosystems in different product domains are gaining an increased amount of attention from the business perspective. Because business models develop dynamically within these product domains, it is challenging for store providers to get a comprehensive overview of the different business models. To help store providers with this overview, we have introduced the Business Variability Model (BVM) to model business model decisions. Based on a modified version of the Orthogonal Variability Model [16] and the Business Model Canvas [19], we develop a three-level (Variation Group, Variation Point, Variant) variability model that simplifies the handling of business model decisions of different stakeholders. The advantage of our Business Variability Model compared to the Business Model Canvas is the possibility to differentiate between mandatory and optional business model decisions and to model dependencies between these decisions. After providing a summarized view of the different design decisions of store providers, we take a deeper look into the different revenue streams and their dependencies to the channels of such software ecosystems. With the results of this paper, the store providers should get an approach to model different business model decisions and receive insights about current existing business models of store-oriented software ecosystems. Both, in turn, will support them by improving their current business situation.

For the future, we have also identified two interesting research directions with deriving of design patterns and the usage of concepts of dynamic software product lines. For deriving of design patterns, we need to model the business model of each software ecosystem using our variability model. Now, we can use pattern mining to identify common patterns of different product domains. As an example of a pattern, the providers of video game consoles combine a closed platform solution together with a subscription service and exclusive brands for customer acquisition. The second point is the usage of concepts of dynamic software product lines. In this paper, we are using the concepts of static software product lines to model the variabilities of the business model. Because business models are highly dynamic it is worth to look if this dynamic can be modeled with them. As an example of this dynamic in the last years, the providers of mobile phone ecosystems started to generate a new revenue stream with the selling of In-Search-Ads.

References

1. Bosch, J., Bosch-Sijtsema, P.: From integration to composition: On the impact of software product lines, global development and ecosystems. Journal of Systems and Software **83**(1), 67–76 (2010)
2. Bouwman, H., Carlsson, C., Carlsson, J., Nikou, S., Sell, A., Walden, P.: How Nokia Failed to Nail the Smartphone Market. In: 25th European Regional Conference of the International Telecommunications Society (ITS) (2014)
3. Chesbrough, H.: Business model innovation: it's not just about technology anymore. Strategy & Leadership **35**(6), 12–17 (2007)

4. E. Porter, M., Heppelmann, J.E.: How Smart, Connected Products Are Transforming Competition. Harvard Business Review **92**(11), 64–88 (2014)
5. Gonçalves, V., Walravens, N., Ballon, P.: "How about an App Store?" Enablers and Constraints in Platform Strategies for Mobile Network Operators. In: Ninth International Conference on Mobile Business and Ninth Global Mobility Roundtable (ICMB-GMR). pp. 66–73. IEEE (2010)
6. Holzer, A., Ondrus, J.: Mobile application market: A developer's perspective. Telematics and Informatics **28**(1), 22–31 (2011)
7. Jansen, S.: Measuring the Health of Open Source Software Ecosystems: Moving Beyond the Scope of Project Health. Information and Software Technology **56**(11), 1508–1519 (2014)
8. Jansen, S., Bloemendal, E.: Defining App Stores: The Role of Curated Marketplaces in Software Ecosystems. In: International Conference of Software Business (ICOB), vol. 150, pp. 195–206. Springer (2013)
9. Jazayeri, B., Platenius, M.C., Engels, G., Kundisch, D.: Features of IT Service Markets: A Systematic Literature Review. In: International Conference on Service-Oriented Computing (ICSOC), vol. 9936, pp. 301–316. Springer (2016)
10. Jazayeri, B., Zimmermann, O., Engels, G., Kundisch, D.: A Variability Model for Store-Oriented Software Ecosystems: An Enterprise Perspective. In: International Conference on Service Oriented Computing (ICSOC), vol. 10601, pp. 573–588. Springer (2017)
11. Jazayeri, B., Zimmermann, O., Engels, G., Küster, J., Kundisch, D., Szopinski, D.: Design Options of Store-Oriented Software Ecosystems: An Investigation of Business Decisions. In: Eight International Symposium on Business Modeling and Software Design (BMSD), vol. 319, pp. 390–400. Springer (2018)
12. La Rosa, M., van der Aalst, W.M.P., Dumas, M., Milani, F.P.: Business Process Variability Modeling. ACM Computing Surveys **50**(1), 1–45 (2017)
13. Lee, S.M., Kim, N.R., Hong, S.G.: Key success factors for mobile app platform activation. Service Business **11**(1), 207–227 (2017)
14. Manikas, K., Hansen, K.M.: Software ecosystems – A systematic literature review. Journal of Systems and Software **86**(5), 1294–1306 (2013)
15. Menychtas, A., Vogel, J., Giessmann, A., Gatzioura, A., Garcia Gomez, S., Moulos, V., Junker, F., Müller, M., Kyriazis, D., Stanoevska-Slabeva, K., Varvarigou, T.: 4CaaSt marketplace: An advanced business environment for trading cloud services. Future Generation Computer Systems **41**, 104–120 (2014)
16. Metzger, A., Pohl, K., Heymans, P., Schobbens, P.Y., Saval, G.: Disambiguating the Documentation of Variability in Software Product Lines: A Separation of Concerns, Formalization and Automated Analysis. In: International Requirements Engineering Conference (RE). pp. 243–253. IEEE (2007)
17. Müller, R.M., Kijl, B., Martens, J.K.J.: A Comparison of Inter-Organizational Business Models of Mobile App Stores: There is more than Open vs. Closed. Journal of theoretical and applied electronic commerce research **6**(2), 13–14 (2011)
18. Nickerson, R.C., Varshney, U., Muntermann, J.: A method for taxonomy development and its application in information systems. European Journal of Information Systems **22**(3), 336–359 (2013)
19. Osterwalder, A., Pigneur, Y.: Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. John Wiley & Sons, Hoboken (2010)
20. Parker, G., van Alstyne, M., Choudary, S.P.: Platform revolution: Platform Revolution: How Networked Markets Are Transforming the Economy - and How to Make Them Work for You. W.W. Norton & Company, New York and London (2016)

21. Roma, P., Ragaglia, D.: Revenue models, in-app purchase, and the app performance: Evidence from Apple's App Store and Google Play. Electronic Commerce Research and Applications **17**, 173–190 (2016)
22. Tuunainen, V.K., Tuunanen, T., Piispanen, J.: Mobile Service Platforms: Comparing Nokia OVI and Apple App Store with the IISIn Model. In: International Conference on Mobile Business (ICMB). pp. 74–83. IEEE (2011)
23. van Gurp, J., Bosch, J., Svahnberg, M.: On the notion of variability in software product lines. In: Conference on Software Architecture (CSA). pp. 45–54. IEEE (2001)
24. Zhang, B., Becker, M., Patzke, T., Sierszecki, K., Savolainen, J.E.: Variability evolution and erosion in industrial product lines. In: 17th International Software Product Line Conference (SPLC). pp. 168–177. ACM (2013)