Subproject C5: Architectural Management of OTF Computing Markets

Gregor Engels[1], Sebastian Gottschalk[1], Dennis Kundisch[2], Christian Vorbohle[2], Nancy V. Wünderlich[3]

1 Department of Computer Science, Paderborn University, Paderborn, Germany
2 Department of Information Systems, Paderborn University, Paderborn, Germany
3 Faculty of Economics and Management, TU Berlin, Berlin, Germany

1 Introduction

Enterprise architectures are used as conceptual blueprints to describe the structure and management of IT systems in organizations. An OTF computer market contains an enterprise architecture that must be continuously aligned with its environment. In times of ever faster-changing environments, architectures must also adapt ever faster with close integration of the business, software, and infrastructure architecture. This section focuses on three aspects of architectural management of OTF computing markets. First, we provide an architectural framework for the static structuring of those markets, including all architecture layers. Second, we focus on the business layer and, in particular, the dynamic business model behavior of the market participants. Third, we identify drivers and barriers to accepting OTF computing markets from multi-stakeholder perspectives. For all aspects, we define the overall research opportunities, show selected highlights of our research, and apply them to the design of OTF computing markets.

The main objective of our research was to understand the design of OTF computing markets from a technical and business perspective. By looking at recent research contributions in the field of enterprise architectures, we recognized that the various research disciplines are more and more intertwined. Here, new substantial contributions result from the structured combination of concepts from existing disciplines. Therefore, we use empirical and conceptual research methods and recombine existing concepts from computer science, information systems, and business administration to develop holistic solutions for OTF computing markets.

As no OTF computing markets currently exist, we conducted our empirical and conceptual studies in comparative markets and discuss how our results apply to the design of future OTF computing markets. Here, we investigated software ecosystems for the architectural framework and business models of single service providers. Next, we discovered business

gregor.engels@uni-paderborn.de (Gregor Engels), sebastian.gottschalk@uni-paderborn.de (Sebastian Gottschalk), dennis.kundisch@wiwi.uni-paderborn.de (Dennis Kundisch), christian.vorbohle@wiwi.uni-paderborn.de (Christian Vorbohle), wuenderlich@tu-berlin.de (Nancy V. Wünderlich)

ecosystems for the interrelationships of different business models. Last, we identified potential drivers and barriers to stakeholders' acceptance of OTF computing markets. Out of that, we derived our corresponding research opportunities.

- **Architecture framework:** Our first research opportunity was to develop an architectural framework for software ecosystems. We first wanted to get an overview of the features of such ecosystems. Based on that, we identified critical design decisions for different architectural layers. Out of that, we derived the main architectural design patterns for various kinds of software ecosystems.
- **Business model development:** Our second research opportunity was to develop business models for ecosystems. We first discovered and analyzed different modeling languages in general and their applicability to business ecosystems. Second, we analyzed different development methods to create a situation-specific business model development approach. Third, we reviewed various software tools for business model development as the foundation for our own one.
- **Attractiveness factors:** Our third research opportunity was to identify success factors for OTF market providers. Based on a literature review on acceptance drivers and barriers from service requesters, service providers, and market provider perspectives, we conducted an exploratory qualitative interview study with potential market participants and market providers.

2 Main Contributions

2.1 Architectural Framework for Software Ecosystems

Nowadays, concerning the changing needs of organizations, simple software solutions have evolved into large-scale software systems [PFG13]. Designing the architecture of those systems in advance is crucial for the success of organizations. However, the designed architectures behind those systems are becoming more complex. Therefore, various architectural approaches are proposed for developing and maintaining enterprise architectures. Among others, the Zachman Framework and the Open Group Architecture Framework (TOGAF) are well-known architecture frameworks for designing software systems. Here, those frameworks structure various layers of the systems, including the business architecture, the software architecture, and the infrastructure architecture. With this, those frameworks aim to provide a closer alignment of the business aspects and technical aspects.

While those architectural frameworks are developed for all kinds of systems, software ecosystems are a subset with special requirements. Here, software ecosystems are defined by Bosch et al. 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*" [BB10]. However, there is an identified lack of reference models for designing ecosystems that support ecosystem providers in their decision-making [AS16]. Therefore, within our research, we identified the architectural design features of those ecosystems and modeled the variability of possible architectural decisions. Out of that, we extracted patterns for various types of software ecosystems.

2.1.1 Features of Software Ecosystems

To support the decision-making in those ecosystems, we first needed to clarify what the common features of those ecosystems are. We have done that by reviewing the literature on IT service markets in a systematic way.

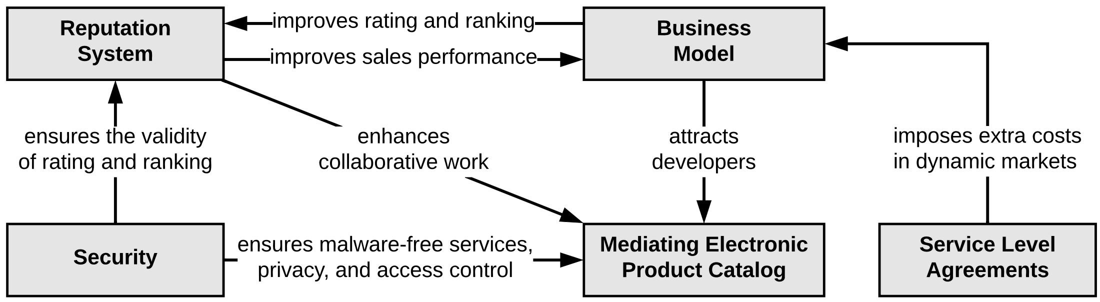

Figure 58: *Primary features of IT service markets [JPEK16].*

The outcomes of our review were the following six primary features, shown in Figure 58, together with several subfeatures [JPEK16]. The *reputation system* is responsible for collecting and aggregating the ratings and reviews of users for the services. Those systems are used to build trust among the different users in the ecosystem and support the ranking of single services. The *business model* is used to describe the rationale of how the ecosystem can create, deliver and capture value for its users. This, in turn, supports the sustainable growth of the ecosystem over time. The *recommendation system* handles the discovery of different services within the ecosystems. With this, the ecosystem ensures the users' acceptance of new services. The *mediating electronic product catalog* acts as an intermediary between the users and the developers to provide the users' access to the services. The ecosystem uses a catalog to support the standardized discovery and delivery of those services. The *security* is used to save the users' privacy and analyze the developers' code. This is important for creating trust among the users for the provided services. The *service level agreements* are used to guarantee the quality of certain services. This ensures the usability of services also in a business context. We created a variability model for designing software ecosystems from those identified sources and initial features.

2.1.2 Design Variabilities for Software Ecosystems

To support the decision-making for new and existing ecosystems, variability models represent alternative design decisions, including various variation points with different variants. To derive such a variability model, we used a systematic taxonomy development method, derived a taxonomy for classifying objects based on their common characteristics, and mapped it to a variability model.

The variability model, shown in Figure 59, consists of three different views based on the architectural layers of enterprise software systems [JZEK17]. For the visualization, we use orthogonal variability models, where mandatory or optional variation points for the layer are connected through a minimum and maximum of mandatory or optional variants as corresponding choices.

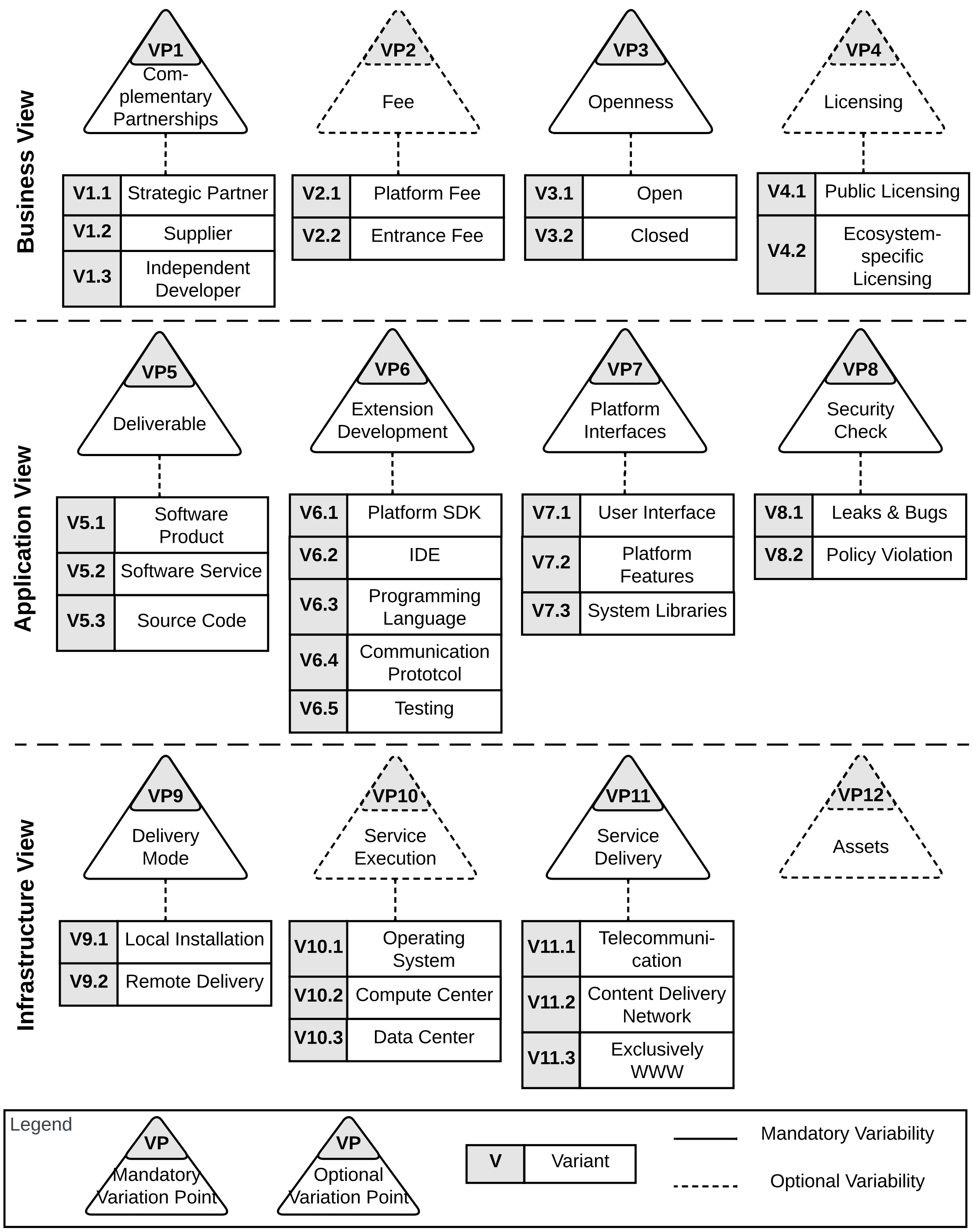

Figure 59: *Variability model of architectural design decision [JZEK17].*

The **Business View** includes the most influential decisions of the business strategy to create a value-capturing environment around the ecosystem. Here, the *complementary partnerships* defines a strategy to choose partners for adding additional services to the ecosystems. *Fees* describe the provision of payments to enter the ecosystem and use corresponding services. *Openness* defines a strategy to which degree access to the ecosystem is possible for the participants. *Licensing* describes how the ecosystem and additional services are licensed to the partners as well as to the participants of the ecosystem.

The **Application View** includes the architectural decisions focusing on the extensibility of the ecosystem by the complementary partners. Here, the *deliverable* provides different types of artifacts for the services the participants could use. *Extension development* provides various techniques to allow partners to develop and test services for the ecosystem. *Platform interfaces* provide various gateways to integrate the developed services into the ecosystem. *Security checks* are integrated into the development or the delivery to protect the participants of the ecosystems from misuse.

The **Infrastructure View** includes the necessary hard- and software to realize the functionalities of the application view. Here, the *deliverable mode* provides different options for delivering the services to the participants. The *service execution* describes the location where those services are actually executed. The *service delivery* shows the backend technology that the ecosystem provider uses to run the ecosystem and deliver the services. The *assets* are additional devices that are used by the ecosystem provider to deliver the ecosystem to the participants. Based on those variabilities, we extract architectural patterns of typical software ecosystems.

2.1.3 Architectural Patterns for Software Ecosystems

To support the decision-making for those ecosystems, patterns describe abstracted knowledge that occurs in multiple organizations. We derive those patterns using a quantitative pattern-extraction method.

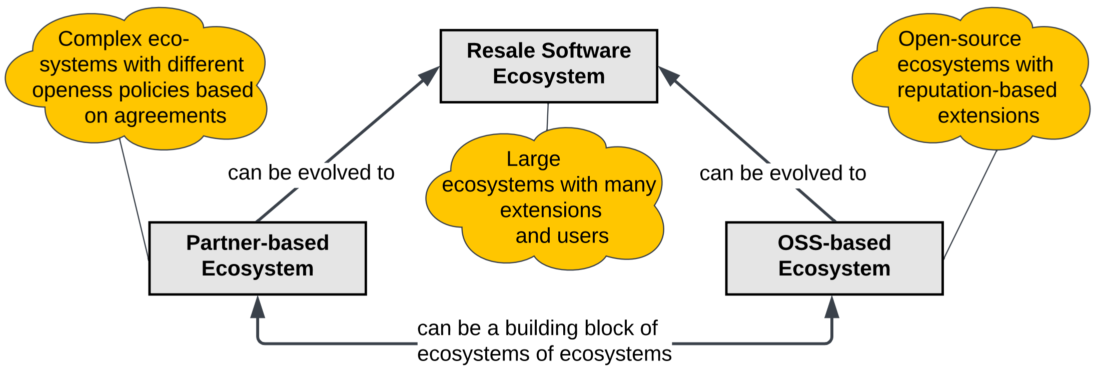

Figure 60: *Architectural patterns of software ecosystems [JZK+18].*

As shown in Figure 60, for the outcome we identified three different patterns of software ecosystems [JZK+18]. Here, *resale ecosystems* provide a large number of extensions by different independent external developers. After their creation, the extensions are sold to many users within the ecosystem. Next, *partner-based ecosystems* are used for complex ecosystems in new industrial sectors, where the external developers and ecosystem providers build new extensions based on partnership agreements. Here, different openness policies support providers in protecting intellectual property within their ecosystems. Last, in *OSS-based ecosystems*, the software platform is mostly released under open source by the ecosystem provider. The external developers are primarily not financially motivated to develop extensions. Instead, they aim to gain reputations or extend the ecosystem for their own purposes. Over that time, the partner-based and OSS-based ecosystems might evolve into resale ecosystems. By considering those results, we developed our architectural framework for software ecosystems.

2.1.4 Application to OTF Computing Markets

Architectural design decisions are essential for ecosystem providers in order to create new or revisit existing ecosystems. However, there was less structured information on the most critical design decision for the different architectural layers of business, software, and infrastructure of software ecosystems available. Therefore, we developed an architectural framework consisting of ecosystem market features, architectural design variabilities for all layers, and possible ecosystem patterns.

Out of that knowledge, we developed an open-source software tool called SecoArc [SE20] as an Eclipse plugin, which contains two main components for the pattern-centric design of software ecosystems. First, ecosystem providers can model the different variabilities of their ecosystems under the consideration of an existing meta-model. Second, the provider can analyze those decisions for conformance errors and receives suggestions for architectural patterns. With the tool, we support ecosystem providers in creating and improving their software ecosystems. Here, we applied our architectural framework to design OTF computing markets. By using the definition of software ecosystems from Bosch et al. [BB10], the OTF market provider might provide the software platform for the ecosystem. In those ecosystems, OTF service providers are the external developers whose services are composed by the OTF providers as domain experts to individual solutions. Those composed services are used by the OTF requestors as users. Lastly, the services are executed in the infrastructure of the OTF compute center. Therefore, we can apply our architecture framework for the market provider to design those ecosystems.

We applied our software tool SecoArc to the OTF Proof of Concept (OTF-PoC). Here, the OTF-PoC[21] is an instance of a potential OTF computing market, where a chatbot interface is used to configure AI-based services. We conducted a case study with the aim of opening the ecosystem for external services by placing representatives of the PoC development team as well as external service providers in the role of potential market providers. The market providers use the SecoArc tool to model different variabilities of the ecosystems. Those variabilities consist of business-related (e.g., free or paid entrance fee (V2.1 in Figure 59)), application-related (e.g., Java and Python as programming languages (V6.3)), and infrastructure-related (e.g., single or multiple servers for remote delivery (V9.2)) design decisions.

Out of those designed decisions, they derived two different architectures. Here the overall vision was to provide "*an ecosystem [that] should support innovation while being sustainable in terms of confronting external threats that could have a long-term impact on the platform's success. Furthermore, the platform ownership should be managed by using the GNU General Public License (GPL).*" The first one is an open ecosystem in which the platform remains an open-source project so that the ecosystem grows through

[21] Website of the OTF-PoC: `https://sfb901.uni-paderborn.de/projects/tools-and-demonstration-systems/tools-from-the-2nd-funding-period/proof-of-concept`

the direct contributions of service providers and the source code can be freely used. This mostly relates to the OSS-based ecosystem architecture design pattern. The second one is a semi-open controlled ecosystem where the number of service providers on the platform drastically increases and the openness needs to be reduced by controlled software development and the marketing environment. This mostly relates to the resale ecosystem architecture design pattern. Based on those architectural patterns, we also need to develop corresponding business models for actors in the business ecosystem. While we were working on the development of the framework and its application to OTF computing markets, we saw a special need for investigating the business models as part of the business architecture.

2.2 Business Model Development for Ecosystems

Business Model Research is a rapidly growing field, and the concept's usefulness has been emphasized in research and practice. For the case of OTF computing, the interplay of business models and technology is especially crucial as "*a mediocre technology pursued within a great business model may be more valuable than a great technology exploited via a mediocre business model*" [Tee10]. Consequently, the concept of business models is a well-established means to offer an essential contribution to the business architecture and acts as an intermediary between business strategy and business processes. A business model describes the design or architecture by which a company creates and delivers value for its customers, thereby generating profit [Tee10]. Some business model definitions also make references to representing a system. By that, business models can also be viewed as a system consisting of activities performed by the company and its ecosystem partners [ZA13]. The challenge in developing business models for IT ecosystems is that every combined product or service can be commercialized via an uncountable number of possible business model alternatives. Developing economically successful business models for all participants in a complex system, such as OTF computing markets, is a substantial challenge and a decisive success factor.

Ecosystem Research is currently receiving increased attention from research and practice. The concept of ecosystems originated in biology and was adapted to the business context in the early 1990s as a community of cooperating companies or individuals. This community creates products and services for customers who are also part of the business ecosystem, such as suppliers, competitors, and other stakeholders. More recent research defines a business ecosystem as companies collaborating to create a focal value proposition [Adn17], consisting of multilateral and non-generic complementarities coupled with the absence of full hierarchical control [JCG18]. This means that what enables businesses to collaborate and align in an ecosystem is creating a shared value proposition for customers. The business ecosystem concept, therefore, calls for a multi-actor assessment of how value is created, delivered, and captured, i.e., an evaluation of whether a viable business model is established for each ecosystem participant.

Both concepts provide the foundation for our second research opportunity: the development of business models for IT ecosystems. However, akin to technology innovation, creating and innovating a business model within an ecosystem environment is a creative and collaborative process. Therefore, starting from the business model and business ecosystem concept, we first analyzed and further developed modeling languages, innovation methods and software tools to overcome knowledge boundaries, depart from traditional business approaches, and utilize innovative ways instead to create, deliver and capture value.

2.2.1 Languages for Business Model Development

Business model modeling languages (BMMLs) explicitly communicate the core logic and elements of a business model and employ "*semantic constructs, visual form, and visual notation to represent the business model of a given organization (but not tied to any specific organization) for one or more purposes and through a consistent set of rules*" [SMJ+22].

Compared to other subfields of research on conceptional modeling (e.g., process modeling), research on business modeling is a relatively young subfield of research and less studied. Given the continuously increasing relevance of business model innovation, we argue that BMMLs will continue to be a relevant research topic for different research disciplines such as computer science, information systems, or strategy. However, research within and across disciplines has remained disparate rather than cumulative. Limited knowledge accumulation is problematic because single contributions tend to remain isolated with little relation to other solutions. The current proliferation of BMMLs substantially aggravates the development of a cumulative research tradition. This underlines the necessity for a firm understanding of the state of the art of modeling languages for business models and a respective research agenda. The primary aim of this identified research opportunity is (I) to advance our understanding of business modeling in general and, more specifically, (II) to determine how companies in business ecosystems can be supported in developing innovative business models collaboratively.

(I) Advance general understanding of business modeling. We observed that general knowledge of BMMLs was limited in at least five major ways: (1) limited consolidation (What BMMLs exist?); (2) limited theoretical grounding (How can we compare different BMMLs?); (3) limited evaluation (What are their similarities and differences?), (4) limited understanding of use (How were they researched and used so far?) and (5) future research opportunities (What do we still need to know?). To answer these questions, we conducted a cross-disciplinary synthesis of widely used BMMLs [SMJ+22].

Limited consolidation. In total, we identified 17 different BMMLs. The most well-known examples are the Business Model Canvas [OP10] and e3value [GA03]. In Figure 61, we demonstrate both modeling languages with an example of a mobile app store. Other examples of BMMLs include the Causal Loop Diagram, ebusiness model schematics, the Strategic Business Model Ontology, and the Value Stream Map.

Limited theoretical grounding. We suggest that BMMLs can be analyzed in terms of three main characteristics: i.e., a) content, b) visual notation and form, and c) context of use, also referred to as semantics, syntax, and pragmatics. The semantics of a modeling language refers to what a language attempts to represent (i.e., the "vocabulary"). Syntax refers to how a modeling language represents content, i.e., the type of visual notation it uses (i.e., graphical symbols) and the type of visual form it takes (i.e., the architectural form of a representation). The pragmatics of a language refers to the context of use under which a modeling language is applied.

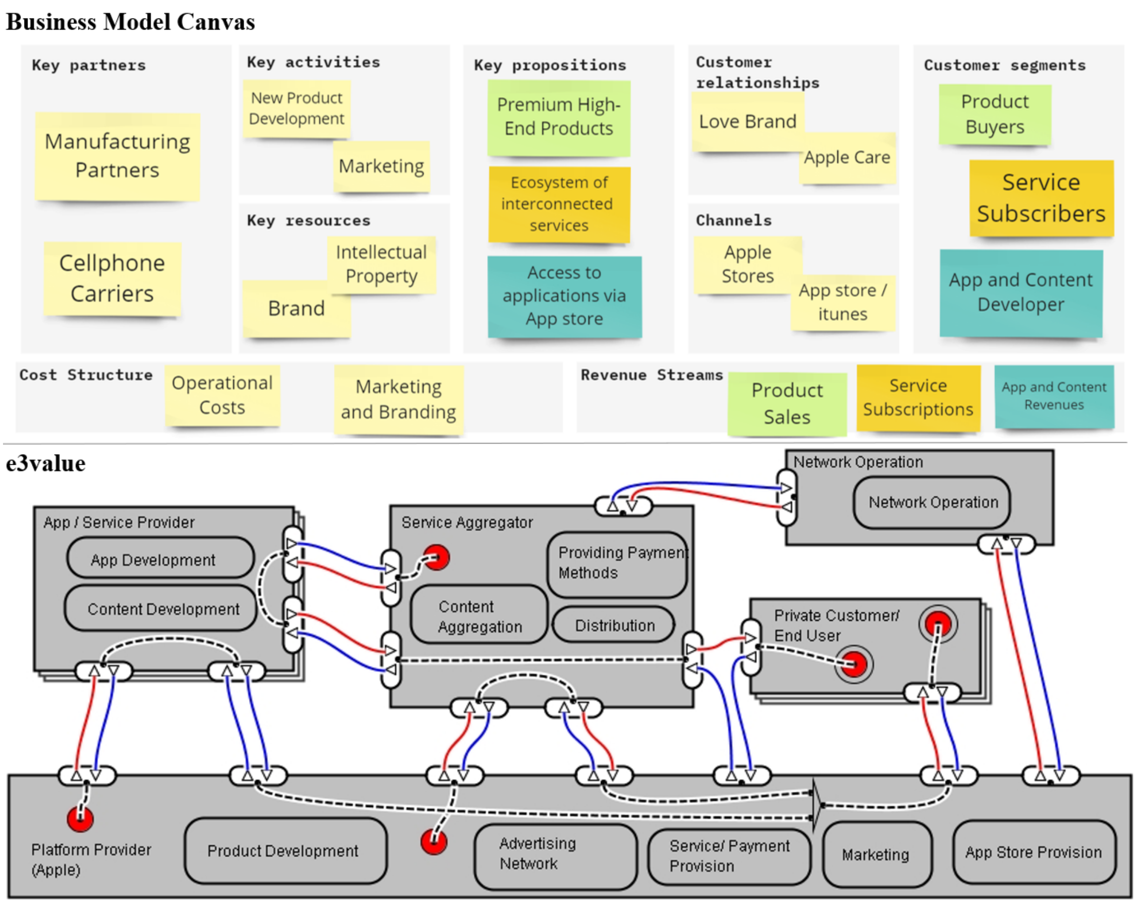

Figure 61: *Two different ways of visualizing a mobile app store business model.*

Limited evaluation. Comparing the identified BMMLs in terms of semantics leads to the identification of different levels (low, moderate, high) of granularity and scope. For example, the Business Model Canvas has a moderate scope and granularity because, with its 9 semantic constructs, it covers 11 semantic sub-dimensions. In contrast, the e3value covers fewer than 9 semantic sub-dimensions and therefore has a lower scope and granularity, just as most other BMMLs. In terms of syntax, the majority use a network-based visualization approach (e.g., e3value), and only three use a map-based approach (e.g. Business Model Canvas). In terms of pragmatics, we identified five main purposes (e.g., generate business model ideas) by analyzing the author's intention to use BMMLs. Here, the intention to use the Business Model Canvas includes all five purposes. In comparison, e3value only includes 4 purposes because it does not intend to support the design of software-based business model development tools. In summary, our identified BMMLs have been developed in a variety of disciplines and for different purposes. No well-accepted set of semantic constructs exists, and various visual notations with a varying number of views for representing semantic constructs have been proposed.

Limited understanding of use. To answer this question, we first analyzed research with BMMLs in greater detail. Summarizing the purposes of employing BMMLs for research, there is no systematic coherence between BMMLs and the purpose for which they are used. However, very different BMMLs are used but for different purposes, partly also within the same research discipline. Nevertheless, we identified four different ways of how the purposes described before are realized with the help of a BMML: (1) artifact development, (2) data collection, (3) business model analysis, and (4) results communication. Regarding research about BMMLs, we identified multiple studies across four research streams. The majority researched the Business Model Canvas, followed by e3Value. Overall, our review reveals eight purposes for conducting research about BMMLs, e.g., for supporting the development of software or design business models for sustainability. Across the eight identified purposes, there are three different ways of how these purposes are pursued to research about BMMLs: studies that (1) link BMMLs with other modeling languages, (2) extend BMMLs with respect to the respective disciplinary background, (3) theoretically ground BMMLs (e.g., [SL20]).

Future research opportunities. We took advantage of the opportunity that this thorough analysis offers to suggest avenues for moving forward. For that, we used the same dimensions to compare BMMLs, namely semantics, syntax, and pragmatics. Future research should bridge existing knowledge and integrate investigations in areas such as creativity and innovation management, information systems, or marketing and strategy. These areas have the potential to contribute to research with and about BMMLs. Moreover, two emerging research directions offer great potential for further investigations namely digitally enabled business models and sustainability. Figure 62 offers an illustration and synthesis.

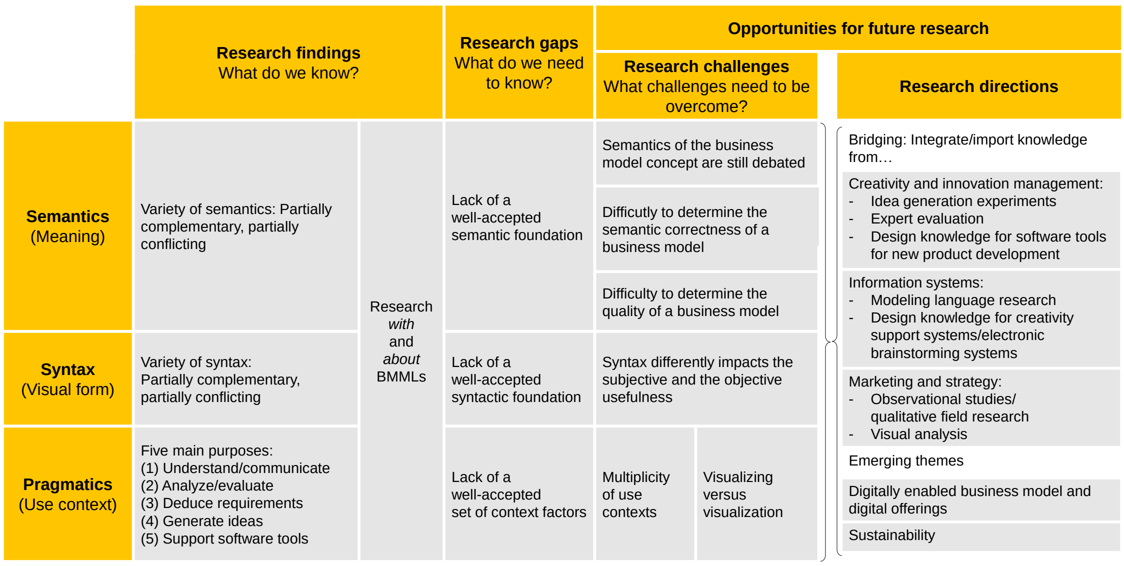

Figure 62: *Summary of research on business model modeling languages [SMJ+22].*

(II) Advance a specific understanding of business modeling for business ecosystems. Based on these findings and identified future research directions, in [VK22] we analyzed the merits and limitations of existing BMMLs to design and analyze business ecosystems. In terms of semantics, we focused on analyzing relationship types to describe different possibilities for connecting business models or business model components (e.g., structural, dependency, dynamic). The multitude of BMMLs appear network-based and represent interactions between actors in a business network. While most BMMLs in this segment are limited to exchanging value streams, others include intangible exchanges such as knowledge, internal impacts and goals, resources, and processes. Moreover, we built on existing ecosystem theory, for example, from [Adn17] and [JCG18], and derived four dimensions for ecosystem analysis: (1) contingency risk, (2) specificity, (3) governance, and (4) dynamics.

Despite network-based BMMLs being strongly related to enterprise interaction, BMMLs need to pay more attention to the visualization of a shared value proposition and are therefore less suited to analyzing the characteristics of companies collaborating in the same business ecosystem. In general, our analysis shows limited possibilities to analyze business ecosystems from the perspective of ecosystem theory.

By advancing both the general and specific understanding of BMMLs we provide a starting point for further systematic comparisons and build on these findings by creating a modeling approach that sets the focal value proposition in the center of the visualization. In this way, we enable business modeling not only for intra-organizational business model innovation but also for inter-organizational alignment, as in the case of OTF computing markets. With our extensive knowledge base on BMMLs, future research holds great potential to enable collaboration, thus contributing to the next steps of business modeling.

2.2.2 Methods for Business Model Development

To develop business models, domain experts in research and practice have proposed various business model development methods (BMDMs) with different levels of detail in their usage. To support the business developer in using these levels, we have conducted a design science research study to propose a situation-specific business model development approach that composes BMDMs to a specific situation of the organization. This situation-specific adaptation has already proven its value in Situational Method Engineering (SME) [HRÅR14], in which situation-specific software development methods are constructed from fragments of a method repository. Our approach, as shown in Figure 63, introduces five roles that are centered around three stages [GYNE22a].

The first stage of **Knowledge Provision of Methods and Models** is used to utilize the knowledge about the development methods of the business model development and the (canvas) models to visualize of the (business) models. In the beginning, the *meta-method engineer* creates meta-models for the repositories of the methods and (canvas) models *(1.1)*. Here, the *method repository* is able to store different development steps together with development phases or development step sequence patterns for later structuring. Moreover, the *(canvas) model repository* is able to store different models of canvasses and templates together with predefined information on them. Next, different *domain experts* explain their domain knowledge of existing development methods and (canvas) modeling artifacts to the *method engineer (1.2)*. The *method engineer*, in turn, formalizes that knowledge in terms of development methods and within (canvas) models according to the meta-models to make it usable within the approach *(1.3)*.

The second stage of **Composition of Development Methods** is used to construct the *development method* out of the *method repository* and link method steps to the different *(canvas) models* in the *(canvas) model repository* that are used within the development. Here, the *business developer* of the organization explains to the *method engineer* the current context in terms of the situation of the organization and application domain of the service in which the business model should be developed *(2.1)*. The *method engineer*

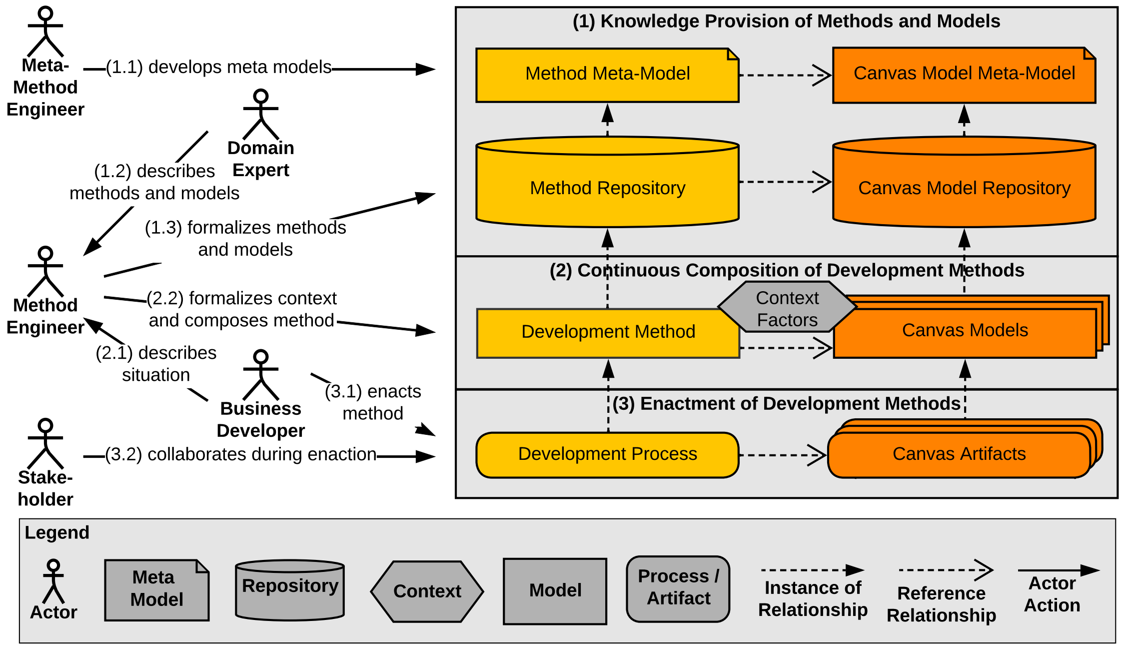

Figure 63: *Situation-specific development of business models [GYNE22a].*

formalizes these *context factors* as the situation of the *development method* and application domain of the *(canvas) models* together with composing the *development method (2.2)*. Here, we support the pattern-based and phase-based composition of development methods. For the pattern-based composition, we select different patterns based on the situation from the *method repository* and nest them into each other. After that, we fill placeholders in those patterns with development steps that are also selected concerning the situation. For the phase-based composition, we select the different phases we want to support from the *method repository*. After that, we select development steps for each phase concerning their situation and order their execution sequence. For both compositions, we connect single development steps to canvas models or template models in the *(canvas) model repository* that are selected concerning the application domain of the service.

The third stage of **Enactment of Development Methods** is used to execute the *development process* and create and modify corresponding *artifacts* during the development. Here, the *business developer* executes the constructed *development method* as a *development process* and uses the linked *(canvas) models* as *(canvas) artifacts* like for the canvasses or templates *(3.1)*. Moreover, other *stakeholders* can contribute to different development steps and modify *(canvas) artifacts* during the execution *(3.2)*. Here, that execution can also lead to a change of the context and, therefore, to a modification of the development method.

We applied our approach in two different domains. In the first domain, we analyzed gray literature to develop business models for mobile applications [GYNE21]. Here, the development methods are structured according to identified method patterns and canvas models are used during the enactment. In the second domain, we analyzed design thinking techniques for conducting design thinking workshops [GYNE22c]. Here, the development methods are structured according to different design thinking phases and predefined whiteboard templates are used during the enactment. To support the situation-specific development of business models, we have implemented the whole approach in a software tool.

2.2.3 Software Tools for Business Model Development

Various software-based business model development tools (BMDTs) have been developed in research and practice to support the development of business models. To get an overview of the functionalities of those BMDTs and identify open research topics, we used a systematic taxonomy development to derive an underlying taxonomy of BMDTs functionalities.

	Dimension	Characteristics						
Modeling	Customization	Add	Divide	Link	Rename	Change arrangement		
Modeling	Development	Element	Element connection	Template				
Modeling	Commenting and Linking	Textual comments on element-level	Textual comments on business model-level	Graphical comments (predefined)	Graphical comments (freedom)	Link files	Link web-resources	Glossary support
Modeling	Assessment	Financial	Non-financial	Assessment status	Correctness checker			
Modeling	Navigation and Filtering	Model comparison	Element filter	Phase management	Element clipboard	Link to business model	Framework support	
Collaboration	Communication	Chat	Discussion board	User list				
Collaboration	Synchronization	Asynchronous modeling	Concurrent modeling	Synchronous modeling				
Collaboration	Assessment	User management	Role management	Support of task sharing	Workspace awareness			
Collaboration	Communication	Version control	Local repository	Remote repository				
Technical	Architecture	Client / Server	Client only	Web-based				
Technical	Data exchange	Export	Import					

Figure 64: *Functionality taxonomy for business model development tools [SSJ+20].*

The first outcome is the **Functionality Taxonomy** with the following three perspectives [SSJ+20] as shown in Figure 64. The *modeling* perspective refers to functions used particularly during the creation of a business model. Those functionalities range from the customization of the underlying models, over commenting and linking on boards, to navigation and the filtering of instances. The *collaboration* perspective presents functions that support collaboration during the business model development. We derive functionalities such as communication through the users, the synchronization of the modeling, or the management of users and roles. The *technical* perspective describes the technical attributes of those tools. It consists of the communication architecture of those BMDTs together with their exchange of data.

The second outcome is a **Future Research Agenda** with the following five topics. This agenda is structured around the groups of future functions, the evaluation of performance, the incorporation of user and task characteristics, and the methods used. Here, the group of methods topics refers to the variety of ways a BMDT for business model development can be used. Here, its usage can be ranged from micro-level (e.g., the decision for a moderating team member) to macro-level processes (e.g., usage of a specific development method) or specific to the reasons for an organization (e.g., developing a new business model vs. improving an existing business model). One open challenge is the composition and enactment of business model development methods according to the situation of the organization.

Based on those functionalities and open research in the method group, we developed the *Situational Business Model Developer (SBMD)* [GYNE22b] as a BMDT, which was derived as an IT artifact from the DSR study for situation-specific business model development [GYNE22a]. It is a web-based solution that can be used within the web browser,[22] and the source code is freely accessible.[23] It supports all the stages of knowledge provision of methods and models, the composition of development methods, and their enactment. The tool can be used by a single user or multiple users can collaborate during the development steps. Moreover, the tool is based on a modular architecture, making it extensible for new methods, models, and development support techniques.

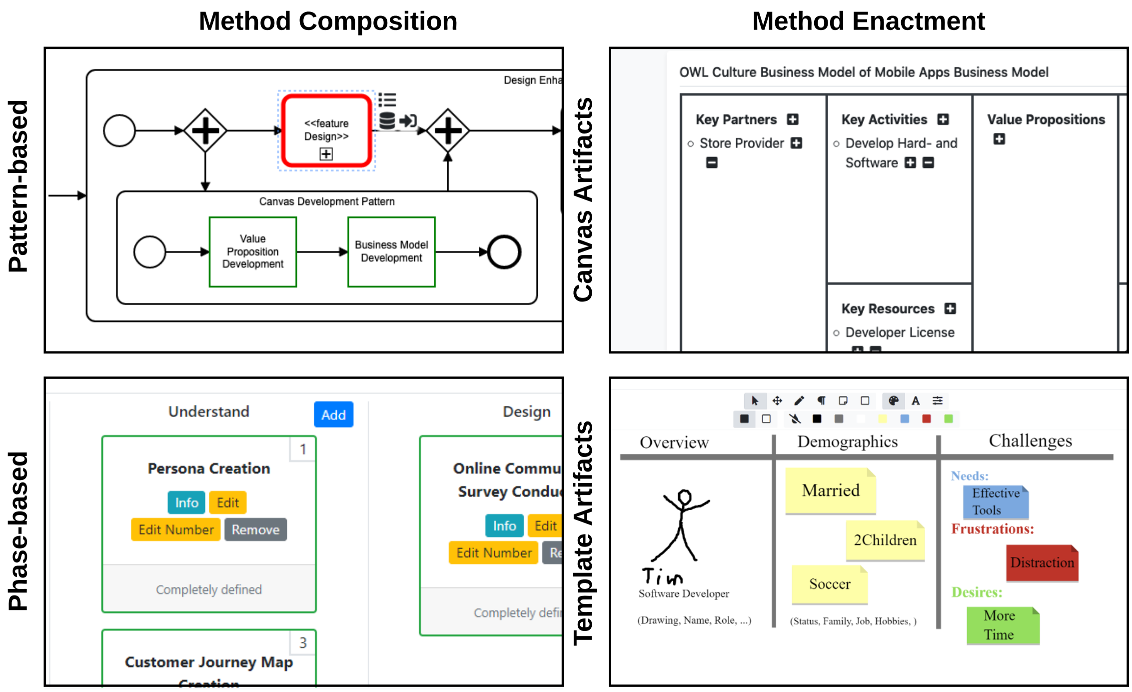

Figure 65: *Screenshots of the Situational Business Model Developer.*

Parts of our **Situational Business Model Developer** for the composition and enactment of development methods can be seen in Figure 65. Before the composition and enactment, our tool provides the knowledge of methods and models by filling the predefined repositories. We filled those repositories with knowledge about business models for mobile apps and workshops for design thinking. Nevertheless, those repositories can be continuously

[22]Online version of the Situational Business Model Developer: `http://sebastiangtts.github.io/situational-business-model-developer/`

[23]Source code of the Situational Business Model Developer: `https://github.com/sebastiangtts/situational-business-model-developer`

extended by the users of the tool. For the method repository, we allow the creation of atomic method elements combined into method building blocks as development steps and optionally structured according to method patterns. For the (canvas) model repository, we allow the creation of atomic (canvas) elements that are structured into (canvas) building blocks and visually represented through (canvas) models. For the *method composition*, and after defining the context, we support using patterns and phases. During the *pattern-based* composition, we support the combination of different method patterns to structure the method building blocks. During the *phase-based* composition, we support selecting different phases from the method elements and assign of multiple method building blocks to them. For the *method enactment*, we support the creation of canvas artifacts and template artifacts, besides from simple text documents. The *canvas artifacts* are visual representations of canvas models where predefined knowledge of the canvas building blocks supports filling out of the boxes. The *template artifacts* are visualizations of the template models, which can be freely filled out. During the enactment, the conduction of development steps might also lead to a change in the context and, therefore, a change in the composed development method. A deeper explanation of the tool is available in the explanation section within the tool. By considering those results, we draw the lessons learned for ecosystem business model development.

2.2.4 Application to OTF Computing Markets

The concept of business models and business ecosystems are well-established means to contribute to creating a business architecture. Using these concepts, we analyzed BMMLs, innovation methods, and software tools in general, identified current research gaps, and proposed a research agenda with future research directions. Building on this research on modeling languages, methods, and tools, we developed a procedure model for business model collaboration, such as in the case of OTF computing markets.

In [RLL+22], we offer a comprehensive new set of methods and modeling approaches, which should be used in a workshop setting with multiple company representatives and can be supported by collaboration tools such as Miro or as a part of design thinking workshops in our SBMD. The BMI4BE procedure model can be used for two possible application contexts: (1) to innovate an already existing business ecosystem or (2) to design a new business ecosystem. By this, the procedure model is also an extension of existing business model innovation methods (e.g., The Business Model Canvas).

The idea of the procedure model (see Figure 66) is based on the consideration that representatives of the individual companies first focus on creating value for potential customers of the business ecosystem to develop a shared value proposition and a market perspective. Subsequently, the (potentially) participating companies' contributions to the ecosystem are specified, and the necessary business model flows are developed. On this basis, a common visualization of the ecosystem is created to analyze the flows between the individual companies. In the last step, every company uses these results to develop the necessary transformation for their current business model. These steps can be repeated iteratively until every representative agrees on the outcome.

Figure 67 shows an example of the visualization of a possible ecosystem flow map (Phase 5) for OTF computing markets. This example was created during a CRC901 research

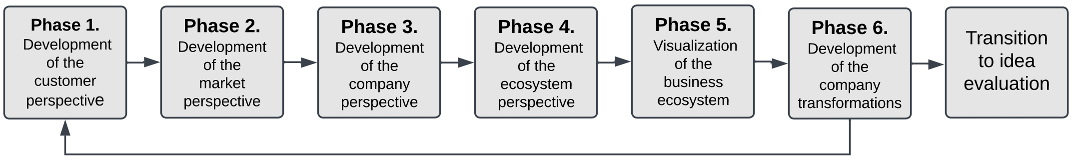

Figure 66: *Phase diagram of the BMI4BE procedure model [RLL+22].*

seminar in May 2022 with researchers from other subprojects and is one of four innovative solutions created during a workshop. The example in Figure 67 shows that OTF computing markets could serve more than one customer segment, which mainly interacts with a market provider. Behind this market provider, OTF computing solutions are generated by an ecosystem of multiple OTF providers, which combine modules from the service provider, infrastructure provider and component provider. Moreover, computing centers deliver fast cloud solutions for small and medium-sized enterprises. As this example shows, the market provider and its attractiveness to potential customers play a central role. These attractiveness factors are further analyzed in the next section.

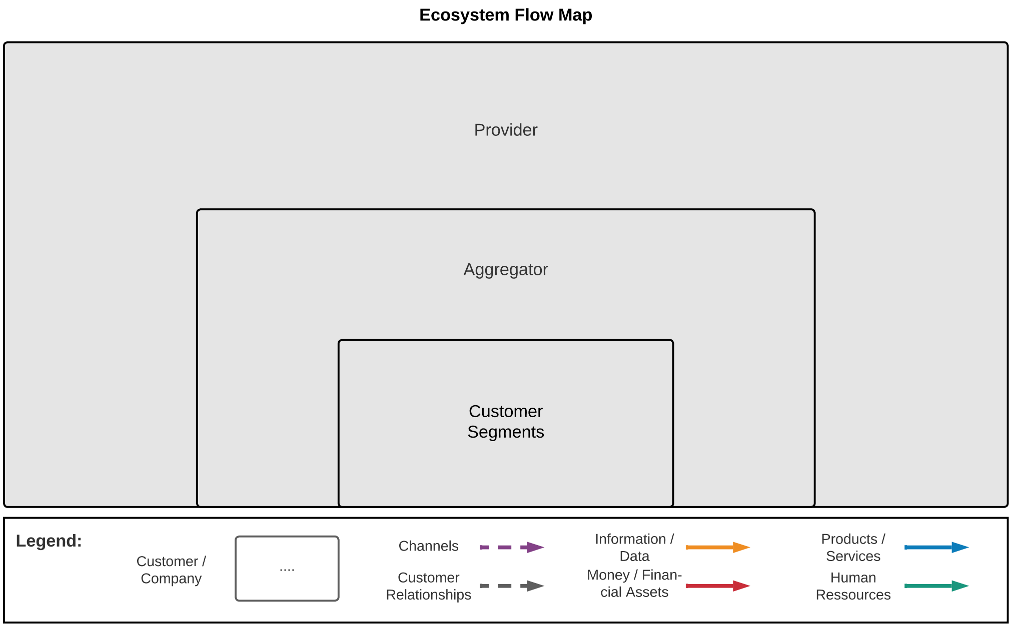

Figure 67: *Ecosystem flow map of an OTF computing market.*

2.3 Attractiveness of Platforms

Platforms can be viewed as particular kinds of markets that play the role of facilitators for an exchange or a transaction between different types of stakeholders that could not otherwise, transact with each other, or only with great difficulty [BKW21]. We define a digital platform as a mediating entity operating in two (or multi)-sided markets, which uses the Internet to enable direct interactions between two or more distinct but interdependent groups of users so as to generate value for at least one of the groups.

In our research, we explore the success factors of digital platforms based on a literature review on drivers and barriers of technology acceptance in related contexts and on an empirical qualitative research study with potential stakeholders of a future OTF market.

2.3.1 Stakeholders of Matchmaking Platforms

Our research focuses on the application of the OTF market as a type of matchmaking platform such as Airbnb and TaskRabbit, which are often characterized by "*high platform intermediation as well as high levels of consociality*" [PK18]. Matchmaker platforms provide the infrastructure to facilitate interactions among the platform users: the service providers on one side and the service requesters/end users on the other. Market providers of such matchmaking platforms typically do not provide the services themselves that are offered on the platform (e.g., only hosts offer accommodation). The core value proposition of a matchmaking platform provider comes from providing a pairing of market participants (e.g., matching hosts with guests [PK18]). Thus, market providers constitute new organizational forms that rely on individual service providers as their co-producers, who are not employees of the platform, but usually act as independent entrepreneurs. On matchmaking platforms, we observe triadic service relationships between three stakeholder entities: the market provider, the service providers, and the service requesters.

Thus, the relationship management measures of a platform provider differ considerably from the relationship management of traditional customer-firm relationships given the unique characteristics of matchmaking platform business models. A market provider not only has to manage a requester base and attract and retain a critical mass of requesters on the platform but also needs a critical mass of service providers that guarantee the service provision and consumption on the platform. Thus, it is crucial for market providers to understand what drives the attraction of matchmaking platforms in order to recruit and retain participants on all market sides.

2.3.2 Framework of Drivers and Barriers of Platform Acceptance

Literature on platform acceptance is sparse and often reflects only one stakeholder perspective (e.g., end user's perspective). For example, [JPML21] conducted a study with 450 Airbnb guests and identified network externalities, trust, perceived ease of use, perceived usefulness, and the level of interactivity on the platform as factors influencing end users' intention to purchase on the Airbnb platform. Since there is no study that brings together the acceptance drivers and barriers from multiple stakeholders, we develop a framework on multi-stakeholder perspectives on the acceptance of matchmaking platforms. Figure 68 illustrates such a framework.

To develop our framework, we draw from different literature streams that explore potential acceptance factors pertaining to one or more perspectives. In particular, we draw on literature on technology acceptance from an end-user and employee perspective (e.g., [Dav89]), literature on the digital transformation of firms (e.g., [WLCH19]), and literature on work-relationships of employees and solo entrepreneurs (e.g., [KBE21]).

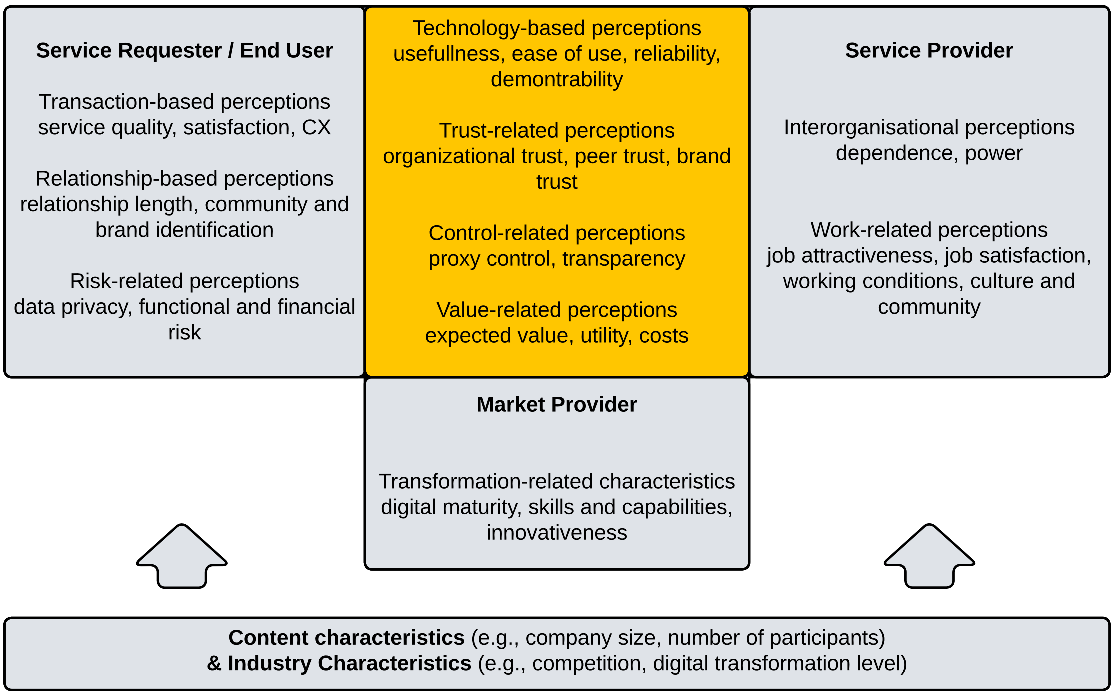

Figure 68: *Framework on multi-stakeholder perspectives on the acceptance of matchmaking platforms.*

Technology perceptions, transcending trust- and control-related beliefs and value perceptions are likely to affect the acceptance of matchmaking platforms for service requesters, service providers and market providers alike. Technology perceptions such as ease of use and perceived usefulness [Dav89] or reliability and result demonstrability have been shown to positively affect the adoption decision of individuals as well as organizational adoption decisions. Trust-related beliefs pertain to trustworthiness perceptions of individuals such as market participants as well as organizational trust in market providers. Trustful relationships have been shown to foster the acceptance of platforms for end consumers and organizations alike. Especially the trust in the brand of the market provider is likely to affect the acceptance of service requesters and service providers. Parallel to trust research, studies on the concept of control have shown that control is a human driving force. Control over technology addresses an individual's or an organization's need to demonstrate competence, superiority, and mastery of technology. Eventually, the value market participants see in using the platform can be considered a main driver for platform acceptance. End requesters might compare the costs and utility of competing platforms to make an informed decision.

In addition to the aforementioned drivers that might pertain to multiple OTF market stakeholders, studies have identified factors that predominantly relate to one stakeholder group. For example, end users'/requesters' acceptance of platforms is likely to be influenced by relational variables beyond trust, such as the length of the relationship between the customer and the platform or individual service providers. Especially in the context of matchmaking platforms with a high level of consociality, a strong platform brand or community identification of end requesters might even outweigh risk perceptions. Research has identified that in technology-mediated service encounters such as transactions on platforms, risk perceptions (e.g., regarding data privacy, and functional or financial risks) are generally pronounced [PW16]. Although the relationship between the service providers and the market provider does not resemble a professional work relationship, service providers on platforms, who act as solo entrepreneurs, might also count in work-related factors in their decision to use the platform [KBE21]. Interorganizational perceptions such as the service providers' perception of dependence on the market provider, their perception of autonomy, and the power the market provider has over the service providers might influence job satisfaction. Moreover, when an adoption decision is perceived as forced, individuals are likely to have negative emotions and increasing switching intentions. Also, typical work-related perceptions such as job attractiveness, strong relationships, a pleasant working condition, and a friendly work culture and community have been shown to impact employee job satisfaction. We argue that such factors can be considered drivers of platform acceptance, especially if the service providers act as solo entrepreneurs.

A company's decision to extend the pipeline business model or switch to a platform business model and become a market provider is typically dependent on organizational characteristics such as capabilities, the innovativeness of the management and staff, and the overall digital maturity of the firm [WLCH19]. In addition, context factors of the market or network, such as the number of participants, company sizes and industry characteristics such as the competitive level and the level of digitization of the market, are likely to strengthen or weaken the importance of the identified acceptance factors from the perspectives of service requesters, service providers, and market providers.

2.3.3 Application to OTF Computing Markets

We conducted a scenario-based qualitative interview study with 22 potential stakeholders from research institutions as well as companies of a future OTF market. Overall, the study yielded text data from 163 pages of transcribed interview material. Based on the text analysis we could support the importance of most drivers and barriers of our framework (see Figure 68). Most importantly, technology-based perceptions seem to be relevant for all perspectives. For example, one interviewee mentioned the ease of use of the platform as a main driver for the service requesters' acceptance: "*Maybe how it's easy to use that platform. And how you get assistance from … our salespeople. … Yeah. I think that's the difference.*" (ID 11). Also, value-based perceptions seem crucial for accepting the OTF market from the service requester and service provider perspective as one interviewee mentions: "*You've got to create some value to motivate the users to contribute.*" (ID 5). Brand-related trust as well as community and brand identification were seen as drivers of service requesters' acceptance: "*And of course, it's very important that this platform has a good image. … the surrounding communication is very important. … What are the comments from users? Is there anything building up? … Do you get positive comments from friendly users?*" (ID 6).

Beyond identified factors that apply to the acceptance of individual market participants, we could support the importance of transformation-related factors such as organizational capabilities that influence a company's decision to transform into a market provider. The necessity to develop and maintain a sophisticated skillset of the staff of the market provider is illustrated by the following quote: "*And you also have to think about crucial factors of developing platforms … I think you have to have an interdisciplinary team of different* *roles. So, you need, someone who's professional a UX design, you need an IT expert, you need someone who understands how the platform works. ... I think a big success factor [is] that you've got this mixed team of different roles, different experts, that all work together. Because it's pretty complicated to build a platform from scratch. ... You have to combine a lot of knowledge*" (ID 5).

In sum, we observe that the attraction factors of OTF computing markets are multi-faceted and apply to three perspectives: market provider, service provider, and service requester. Managers of OTF computing markets are encouraged to invest equally in the attraction and retention of both market sides.

3 Impact and Outlook

OTF computing markets are a promising new type of ecosystem to increase the value for potential end users by combining the offering from all involved market participants. In our research, we investigated the underlying enterprise architecture, the development of business models, and acceptance factors for the platform itself as well as the market participants. Our results impact the design of future OTF computing markets as well as the analyzed comparative markets such as mobile ecosystems. For that, first, our developed architectural framework and the SeCoArc tool support software ecosystem designers in the analysis of existing software ecosystems as well as the design of new ones. Second, our BMI4BE method is based on multiple research results from prior research and is an interdisciplinary approach developed by researchers from business administration, information systems, and computer science. By that, the method is one of the first considering multiple research disciplines and was communicated to practice in [RLL+22]. Third, our SBMD tool has been developed for the comparative market of mobile ecosystems. The SBMD is released under open source so that it can be extended by additional features, for example, crowd validation. Fourth, we identified factors that impact the attractiveness of OTF computing markets for multiple stakeholders. These results, for example, that end users' acceptance of platforms is likely to be influenced by the length of the relationship between the customers and the platform provides a solid foundation for further investigations for successful and sustainable OTF computing markets.

In the future, we aim to increase the impact of our contributions by research on simulation and evolution. For that, our first aspect covers dynamics in software ecosystems. Here, we want to understand the evolution of those ecosystems over time with their business, technical, and infrastructure aspects. Second, we want to research the simulation of business models [KV23]. Here, we plan to calculate the financial assessments of business models under different internal and external circumstances. Third, we want to investigate the changes in stakeholder perceptions over time. Here, the aim is to understand changes in the short-term, mid-term, and long-term relationships among the stakeholders.

Acknowledgments

We would like to thank Bahar Schwichtenberg, Daniel Szopinski, and Helene Rebelo for their contributions to Subproject C5 over the last years.

Bibliography

[Adn17] ADNER, R.: Ecosystem as Structure. In: *Journal of Management* 43 (2017), no. 1, pp. 39–58

[AS16] AXELSSON, J.; SKOGLUND, M.: Quality assurance in software ecosystems: A systematic literature mapping and research agenda. In: *Journal of Systems and Software* 114 (2016), pp. 69–81

[BB10] BOSCH, J.; BOSCH-SIJTSEMA, P.: From integration to composition: On the impact of software product lines, global development and ecosystems. In: *Journal of Systems and Software* 83 (2010), no. 1, pp. 67–76

[BKW21] BEVERUNGEN, D.; KUNDISCH, D.; WÜNDERLICH, N.: Transforming into a platform provider: strategic options for industrial smart service providers. In: *Journal of Service Management* 32 (2021), no. 4, pp. 507–532

[Dav89] DAVIS, F. D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. In: *MIS Quarterly* 13 (1989), no. 3, p. 319

[GA03] GORDIJN, J.; AKKERMANS, J. M.: Value-based requirements engineering: exploring innovative e-commerce ideas. In: *Requirements Engineering* 8 (2003), no. 2, pp. 114–134

[GYNE21] GOTTSCHALK, S.; YIGITBAS, E.; NOWOSAD, A.; ENGELS, G.: Situation-Specific Business Model Development Methods for Mobile App Developers. In: *Enterprise, Business-Process and Information Systems Modeling*. Ed. by AUGUSTO, A.; GILL, A.; NURCAN, S.; REINHARTZ-BERGER, I.; SCHMIDT, R.; ZDRAVKOVIC, J. Vol. 421. Lecture Notes in Business Information Processing. Cham: Springer International Publishing, 2021, pp. 262–276

[GYNE22a] GOTTSCHALK, S.; YIGITBAS, E.; NOWOSAD, A.; ENGELS, G.: Continuous situation-specific development of business models: knowledge provision, method composition, and method enactment. In: *Software and Systems Modeling* (2022), no. 22, pp. 47–73

[GYNE22b] GOTTSCHALK, S.; YIGITBAS, E.; NOWOSAD, A.; ENGELS, G.: Situational Business Model Developer: A Tool-support for Situation-specific Business Model Development. In: *Wirtschaftsinformatik 2022 Proceedings*. AIS, 2022

[GYNE22c] GOTTSCHALK, S.; YIGITBAS, E.; NOWOSAD, A.; ENGELS, G.: Towards Software Support for Situation-Specific Cross-Organizational Design Thinking Processes. In: IWSiB ’22. Pittsburgh, Pennsylvania: Association for Computing Machinery, 2022, pp. 1–8.

[HRÅR14] HENDERSON-SELLERS, B.; RALYTÉ, J.; ÅGERFALK, P. J.; ROSSI, M.: *Situational Method Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014

[JCG18] JACOBIDES, M. G.; CENNAMO, C.; GAWER, A.: Towards a theory of ecosystems. In: *Strategic Management Journal* 39 (2018), no. 8, pp. 2255–2276

[JPEK16] JAZAYERI, B.; PLATENIUS, M. C.; ENGELS, G.; KUNDISCH, D.: Features of IT Service Markets: A Systematic Literature Review. In: *Service-Oriented Computing*. Ed. by SHENG, Q. Z.; STROULIA, E.; TATA, S.; BHIRI, S. Vol. 9936. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 301–316

[JPML21] JUNG, J.; PARK, E.; MOON, J.; LEE, W. S.: Exploration of Sharing Accommodation Platform Airbnb Using an Extended Technology Acceptance Model. In: *Sustainability* 13 (2021), no. 3, p. 1185

[JZEK17] JAZAYERI, B.; ZIMMERMANN, O.; ENGELS, G.; KUNDISCH, D.: A Variability Model for Store-Oriented Software Ecosystems: An Enterprise Perspective. In: *Service-Oriented Computing*. Ed. by MAXIMILIEN, M.; VALLECILLO, A.; WANG, J.; ORIOL, M. Vol. 10601. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 573–588

[JZK+18] JAZAYERI, B.; ZIMMERMANN, O.; KÜSTER, J.; ENGELS, G.; SZOPINSKI, D.; KUNDISCH, D.: Patterns of Store-orieted Software Ecosystems: Detection, Classification, and Analysis of Design Options. In: *Proceedings of Latin American Conference on Pattern Languages of Programming*. 2018

[KBE21] Karanović, J.; Berends, H.; Engel, Y.: Regulated Dependence: Platform Workers' Responses to New Forms of Organizing. In: *Journal of Management Studies* 58 (2021), no. 4, pp. 1070–1106

[KV23] Ksouri-Gerwien, C.; Vorbohle, C.: Supporting Business Model Decision-making in B2B Ecosystems: A Framework for Using System Dynamics. In: *Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS)*. 2023

[OP10] Osterwalder, A.; Pigneur, Y.: *Business model generation: a handbook for visionaries, game changers, and challengers*. Hoboken, New Jersey: John Wiley & Sons, 2010

[PFG13] Pernstål, J.; Feldt, R.; Gorschek, T.: The lean gap: A review of lean approaches to large-scale software systems development. In: *Journal of Systems and Software* 86 (2013), no. 11, pp. 2797–2821

[PK18] Perren, R.; Kozinets, R. V.: Lateral Exchange Markets: How Social Platforms Operate in a Networked Economy. In: *Journal of Marketing* 82 (2018), no. 1, pp. 20–36

[PW16] Paluch, S.; Wünderlich, N. V.: Contrasting risk perceptions of technology-based service innovations in inter-organizational settings. In: *Journal of Business Research* 69 (2016), no. 7, pp. 2424–2431

[RLL+22] Robra-Bissantz, S.; Lattemann, C.; Laue, R.; Leonhard-Pfleger, R.; Wagner, L.; Gerundt, O.; Schlimbach, R.; Baumann, S.; Vorbohle, C.; Gottschalk, S.; Kundisch, D.; Engels, G.; Wünderlich, N.; Nissen, V.; Lohrenz, L.; Michalke, S.: Methoden zum Design digitaler Plattformen, Geschäftsmodelle und Service-Ökosysteme. In: *HMD Praxis der Wirtschaftsinformatik* 59 (2022), no. 5, pp. 1227–1257

[SE20] Schwichtenberg, B.; Engels, G.: SecoArc: A Framework for Architecting Healthy Software Ecosystems. In: *Software Architecture*. Ed. by Muccini, H.; Avgeriou, P.; Buhnova, B.; Camara, J.; Caporuscio, M.; Franzago, M.; Koziolek, A.; Scandurra, P.; Trubiani, C.; Weyns, D.; Zdun, U. Vol. 1269. Communications in Computer and Information Science. Cham: Springer International Publishing, 2020, pp. 95–106

[SL20] Schwarz, J. S.; Legner, C.: Business model tools at the boundary: exploring communities of practice and knowledge boundaries in business model innovation. In: *Electronic Markets* 30 (2020), no. 3, pp. 421–445

[SMJ+22] Szopinski, D.; Massa, L.; John, T.; Kundisch, D.; Tucci, C. L.: Modeling Business Models: A cross-disciplinary Analysis of Business Model Modeling Languages and Directions for Future Research. In: *Communications of the Association for Information Systems* (2022), no. 51, pp–pp

[SSJ+20] Szopinski, D.; Schoormann, T.; John, T.; Knackstedt, R.; Kundisch, D.: Software tools for business model innovation: current state and future challenges. In: *Electronic Markets* 30 (2020), no. 3, pp. 469–494

[Tee10] Teece, D. J.: Business Models, Business Strategy and Innovation. In: *Long Range Planning* 43 (2010), no. 2/3, pp. 172–194.

[VK22] Vorbohle, C.; Kundisch, D.: Overcoming Silos: A Review of Business Model Modeling Languages for Business Ecosystems. In: *Proceedings of the 30th European Conference on Information Systems (ECIS), Research-in-Progress*. 2022

[WLCH19] Williams, P. A.; Lovelock, B.; Cabarrus, T.; Harvey, M.: Improving Digital Hospital Transformation: Development of an Outcomes-Based Infrastructure Maturity Assessment Framework. In: *JMIR medical informatics* 7 (2019), no. 1

[ZA13] Zott, C.; Amit, R.: The business model: A theoretically anchored robust construct for strategic analysis. In: *Strategic Organization* 11 (2013), no. 4, pp. 403–411