Modeling Platform Ecosystems*

Tobias Pauli, Emanuel Marx, Sebastian Dunzer, and Martin Matzner

Friedrich-Alexander-Universität Erlangen-Nürnberg, Fürther Straße 248, Germany
{tobias.pauli, emanuel.marx, sebastian.dunzer, martin.matzner}@fau.de

Abstract. Platforms facilitate the creation of complementary modules by third parties and act as intermediaries between different groups of actors. Due to the high degree of collaboration between actors, ecosystems evolve around such platforms. As a result of digitalization, platform business models are becoming viable in more and more domains. Despite the increasing variety of different platform ecosystems, means to clearly specify them are scarce. This conceptual ambiguity impedes comparability of research and knowledge accumulation. To solve this problem, we propose a domain-specific platform ecosystems modeling language (PEML) which builds on seminal platform ecosystem literature. We demonstrate PEML by modeling two real-life platform ecosystems based on online case studies. Our results support researchers and practitioners alike in clearly specifying platform ecosystems.

Keywords: Platform · Ecosystem · Domain-Specific Modeling Language

1 Introduction

Over the course of the last few years, platform firms have dominated the economy in many branches, evident from the examples of Amazon in the retail sector or Airbnb in the hotel industry. A significant factor in the success of platforms is their ability to leverage an ecosystem of actors who contribute to the platform in various ways.

Although all platforms share common characteristics, they still seem rather heterogeneous. Considering and understanding their differences and "complexity is essential in balancing the need on one hand to render platforms a researchable unit of analysis while on the other hand avoiding oversimplification of the phenomenon" [32, p. 4625]. The same holds true for their surrounding ecosystems.

Despite the multitude of platform and platform ecosystem research papers, we miss methods to clearly describe and distinguish platform ecosystems. Consequently, researchers refer to platform ecosystems as a rather vague concept without clearly specifying their unit of analysis. The resulting conceptual ambiguity impedes comparability of research and knowledge accumulation [24]. Additionally, as the "platformization" affects more and more areas that have previously been dominated by non-platform business models [23], practitioners need tools

* Supported by the German Federal Ministry of Education and Research (FKZ 01|S17045) as part of the Software Campus project "I4.oT"

to capture the peculiarities of platform ecosystems in their domain. This is necessary to assess established insights from other domains. Thereafter, practitioners can decide whether these might prove applicable in their case.

Conceptual modeling is well-suited to reduce the physical and social world's complexity by accurately describing instances of real-world phenomena. In related disciplines, modeling facilitated general comprehension of, e. g., value creation (e^3value), business processes (BPMN), systems (SysML), and software (UML). Even though these modeling languages aid in defining facets of a platform ecosystem, they do not cover platform-specific aspects. While cross-organizational multi-level modeling can depict an ecosystem in a very detailed manner, we argue for the necessity of high-level comprehensive modeling of platform-ecosystem specifics to facilitate comprehensibility for non-experts in modeling.

Therefore, in the paper at hand, we strive to provide an answer to the following research question: *How can we create comprehensive conceptual models for platform ecosystems to achieve general consensus about their structure, actors and functionality?* To this end, using Frank's [8] method and guidelines for designing domain specific modeling languages, we create the platform ecosystems modeling language (PEML).

The remainder of this paper is structured as follows. In Section 2, we provide the theoretical foundations regarding platform ecosystems on which we base our modeling language. Section 3 describes our methodological approach to designing the modeling language. Afterwards, Section 4 presents the modeling language. In Section 5, we demonstrate the usage of the PEML by modeling two platform ecosystems from different domains. Section 6 frames the modeling language in its related works. Finally, Section 7 rounds off the paper by outlining both the contributions and limitations.

2 Platform Ecosystems

The concept of platforms originates in the literature on internal product families and has since moved on towards a broader understanding as external industry platforms [9, 31]. We focus on these external platforms, as they lead to the emergence of ecosystems in two ways. From an architectural perspective, platforms can be defined as "products, services, or technologies that [...] provide the foundation upon which outside firms [...] can develop their own complementary products, technologies, or services" [9, p.418]. From a market perspective, platforms represent two- or multi-sided marketplaces by facilitating transactions between different groups of actors [25].

The resulting collection of actors who use the platform as technological basis or marketplace forms the platform ecosystem [24]. This platform ecosystem as the collection of actors "affiliated" with each other through the common platform corresponds to Adner's [1] platform-as-affiliation perspective. However, based on the activity-centric structure perspective, ecosystems can also be defined based on the activities that contribute to a specific value proposition [1]. Consequently, the ecosystem-as-structure perspective stresses the necessary alignment of actors and their activities to create value.

Actors in platform ecosystems comprise complementors, users and a platform owner. In contrast to classic suppliers who enable a product's creation, complementors enable a product's use [2]. In the context of platforms, the complementors provide "complementary" solutions to and via the platform to the users. To facilitate the development of modules, platform owners provide boundary resources such as application programming interfaces (APIs) [10]. At the same time, boundary resources serve as a governance instrument to exercise control over complement quality.

Based on the characteristics described above, a variety of different platforms can be distinguished. Some platforms enable the creation of complementary modules (e.g. Sony's PlayStation), while others merely act as a marketplace (e.g. Airbnb). Many platforms provide both of these functionalities (e.g. iOS and the App Store).

These differences also bear implications for the respective platform ecosystems. The ecosystems around market platforms might resemble an ecosystem as affiliation, whereas technology platforms predominantly cause the evolution of an ecosystem as structure around their specific complementary solutions. The creation of applications on mobile platforms (e.g. iOS) might not be subject to specific complementarities between actors. However, the development of more complex technological solutions on digital industrial platforms (e.g. MindSphere) requires close alignment of many actors in different roles.

Some platforms acting as a marketplace (e.g. Airbnb) will be primarily driven by positive cross-side network effects. These occur if the value of the platform for a certain group of actors increases as the number of actors on the other side grows [17]. The more apartments there are available at Airbnb, the more prospective guests will use the platform and vice versa. On the contrary, other platforms (e.g. Facebook) are built on the premise of same-side network effects: A higher number of facebook users increases the platform's value for other users. As the heterogeneity of platforms grows, providing means to model and distinguish different platform ecosystems is therefore critical for a meaningful analysis.

3 Method

For the development of PEML we follow the domain-specific modeling language engineering method proposed by Frank [8]. The purpose of PEML is to provide means for the comprehensive description of different kinds of platform ecosystems. The scope thereby includes all types of platform ecosystems. However, we mainly focus on digital platforms as they currently are the most relevant type in research and practice.

PEML consists of an abstract syntax describing the semantics of the modeling language, and a concrete syntax comprising the visual notation [20, 26]. Additionally, we provide instructions to simplify entry into modelling and to enable comparison of different models. To create the abstract syntax we extracted key concepts from seminal literature on platforms and ecosystems. As requirements such as ontological completeness and clarity may be relaxed on purpose for domain-specific modeling languages, we do not refer to a specific ontology as a basis [8]. The concrete syntax is an important part of any modeling language, as humans develop and interpret models via graphical and textual elements [8]. Thus, creating a cognitively meaningful visual notation is critical to facilitate communication and problem solving in the later use of a modeling language [19]. To ensure this, we stick to the design principles in [19] and [8] to develop the concrete syntax for PEML. Throughout the development process, we evaluated both the abstract and concrete syntax iteratively. To this end, we modeled different scenarios. Every time we realized that a real-world concept could not be represented by PEML or if discussions arose that indicated ambiguity of chosen concepts, we went back and refined our language.

Consequently, to demonstrate the applicability and contribution of PEML, we show two different platform ecosystem models based on online case studies. The goal was not necessarily to create exhaustive models, but rather to present the application of PEML. First, we modeled the Airbnb platform, as its high degree of popularity and relatively simple structures make it ideal for demonstrating the intuitive use of PEML. Second, we modeled the industrial internet of things (IIoT) platform Siemens MindSphere to illustrate PEML in the context of highly diverse business to business (B2B) ecosystems with a multitude of platform components, resulting in high complexity.

4 Platform Ecosystems Modeling Language (PEML)

4.1 Abstract Syntax

With the development of PEML, we aim for a comprehensive depiction of platform ecosystems while at the same time limiting the number of used concepts. The rationale behind this is to ensure familiarity of a wide range of users with the included concepts [8]. To this end, based on the understanding of platforms presented in section 2, we identified key concepts from platform and ecosystem literature that need to be reflected in PEML. In the following lines, we describe the selected concepts and their semantics.

Actor. Actors are parties that contribute resources and activities to value creation. There are three types of actors in platform ecosystems: platform owners, complementors and users [16]. The platform owner is the actor who provides the platform around which the ecosystem emerges [16]. The owner governs the platform, provides interfaces and rules to complementors and manages the exchange of services. Complementors are actors who offer compatible modules that contribute to the value creation of the focal offer [16]. Actors that consume a platform's value proposition are users. Both users and complementors can either be businesses or consumers. While a real actor entity can operate in different functions on a platform, we focus on their current function in the platform-ecosystem. A traveler using AirBnB to find accommodation, for example, may at the other side of the platform also offer accommodation to others. For PEML, we separate the actual actors from their functions, so that actors only exist as their function in the ecosystem in PEML.

Role. Actors can perform different roles. In PEML, role is a defining construct for a set of actors who possess similar resources or perform similar activities with regard to value creation and capture in relation to the platform. Thus, role is an attribute of actors.

Side. From a market perspective, platforms are two- or multi-sided markets [25]. Accordingly, in the simplest form, sides structure the platform ecosystem based on supply and demand [33]. However, in the case of multi-sided platforms a more differentiated consideration of sides is necessary. YouTube, for example, has a content provider side, a user side, and an advertiser side, besides others. Thus, PEML defines the sides of a platform ecosystem dependent on each actors' links to the platform's high-level value proposition. Hence, sides are attributes of the whole platform ecosystem, whereas roles are actors' attributes.

Value Proposition. The value proposition describes a bundle of products and services that creates value for the customer by satisfying a need or solving a problem [22]. As most complex platforms can not be reduced to a single value proposition, PEML distinguishes two levels of value propositions. On a high level, a platform provides a main value proposition to its customers. For instance, in the case of digital industrial platforms such as Siemens MindSphere, a high-level value proposition would be the connection of products, plants, systems, and machines to harness the wealth of data. Value propositions on a lower level directed at individual customers or customer segments are the defining element for ecosystem boundaries from an ecosystem-as-structure perspective [1]. In digital industrial platforms, such a value proposition could be a specific predictive maintenance solution. Such value propositions are the focal point of sub-ecosystems forming in the overall platform ecosystem.

Link. Actors are connected through links. More specifically, links describe transfers of resources and activities between actors. The specific composition of such transfers may vary and comprise, e.g., material, information, value, funds, or even influence [1]. Additionally, the links relate actors to the value proposition by specifying what they contribute to which value proposition.

Activity. Based on Kapoor [16, p. 6] we define activities as "tasks that underlie the different offers that contribute to the focal offer's user value proposition". Simply spoken, this includes all actions performed by actors that directly or indirectly contribute to the creation of a value proposition on the platform.

Resource. Resources in the context of PEML comprise assets, funds, technologies, intellectual property and other means required for the creation of a certain value proposition on the platform.

Boundary Resource. Independent of resources in a general sense, boundary resources (BR) are interfaces of various kinds that enable interaction of complementors with the platform [10]. Bianco et al. [6] distinguish between application BR, development BR and social BR. Application BR such as application programming interfaces enable the interaction of third party modules with the platform. Development BR such as software development kits support the development of complementary solutions. Social BR foster (knowledge) exchange between ecosystem participants, for example in the form of partner programs or workshops.

Leverage. Leverage describes how actors benefit from participating in the platform ecosystem in the sense of "exercising an influence that is disproportionate to one's size" [31, p. 206]. There are three types of leverage: production leverage, innovation leverage, and transaction leverage [31]. Production leverage refers to economies of scale achieved through the re-use of standardized components. Innovation leverage describes the use of the platform as a basis to create new solutions. The platform's market intermediary character enables the transaction leverage, as it simplifies and fosters transactions among actors. In the context of PEML, leverage is an attribute that characterizes the association between an actor's activities and the platform.

4.2 Concrete Syntax

The design of the concrete syntax was guided by the principles for constructing visual notations proposed by Moody et al. [19] and Frank [8]. In line with Lusch et al. [18], we split the elements in the concrete syntax into the three categories ecosystem, platform, and value proposition. The Figures 1-3 show the concrete syntax elements of PEML.

Ecosystem. The ecosystem modeling elements in Figure 1 comprise actors and links. Actors themselves consist of a role, their resources and the activities that they contribute to the platform ecosystem. Actors can either be *single-instance* or *multi-instance* entities (cf. 1a and 1b). Actors who contribute similar activities and resources to the ecosystem constitute one multi-instance actor. While single-instance actors are labeled with their name, multi-instance actors are labeled with their collective role. As activities and resources are provided by actors, they are always attached to an actor.

Links specify the connections between actors and other entities. To improve the visual feedback of platform interactions, PEML introduces curved edges as *platform-actor links* and straight edges as *actor-platform links* (cf. 1c and 1d). The distinction of the link type depends on the active entity in a transaction.

Further, we distinguish peripheral links from the platform-related links by adding the visual cue of dashed arrows. Peripheral links are those links that have an indirect influence on the platform. Hence, peripheral links display relations in the ecosystem surrounding a platform. Comment boxes (cf. 1f) enable

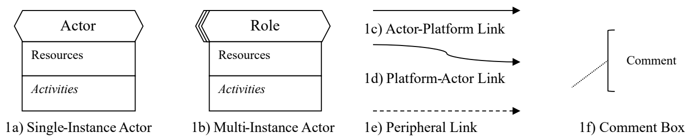

Fig. 1. 1) Ecosystem elements

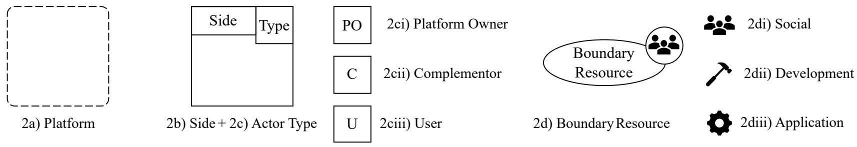

Fig. 2. 2) Platform elements

the modeler to make comments when necessary, especially to clarify the links' composition of transfer.

Platform. The platform elements, depicted in Fig. 2, constitute further syntax elements of PEML. The platform itself (cf. 2a) is a rectangle with rounded corners. It wraps around value propositions (cf. 2d) and boundary resources. A platform generally arbitrates between different groups of actors. These groups of actors are arranged around the platform in the form of sides. PEML depicts *sides* (cf. 2b) as rectangular solid wrappers that contain all actors belonging to the particular side. As a reminder, sides are different from roles, as they classify actors according to their position in relation to the platform, not based on a similar set of activities and resources. Optionally, sides can be labeled, if further clarification is necessary. However, the specification of the actor types within the side is mandatory. Actor types are specified at the level of sides instead of actors, as all actors belonging to the same side always possess the same type. In the square at the top left corner modelers may either add a *U* for user, *C* for complementor, or *PO* for platform owner (cf. 2c). In some cases, complementors and users may be the same real-world entities. At the same time, platform owners might also act as complementors. We propose to split these real world entities into their respective types and assign them to several sides if necessary for the modeling with PEML.

Value proposition. Figure 3 presents all symbols that relate to the value proposition when modeling platform ecosystems. The value proposition is a box with rounded corners (cf. 3a, 3b) with leverage ports (3c). The modeling of functions in control engineering inspires our approach to model value propositions. The value proposition receives at least one input and delivers at least one output. Inputs are specified using the platform-actor and actor-platform links. PEML emphasizes the platform's general value proposition via bold borders and text. The leverage ports define which kind of in- or output the value proposition provides. A leverage port can contain either innovation, production, or transaction

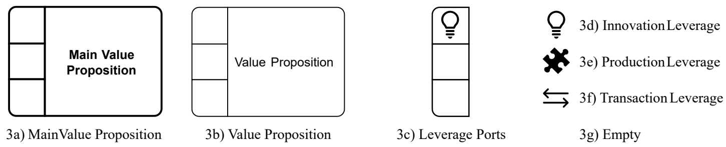

Fig. 3. 3) Value proposition elements

leverage (cf. 3d-3f). If the value proposition does not build on leverage by or provide leverage for any of its participants, the leverage ports remain empty.

4.3 Instructions

The complexity of platform ecosystems usually takes on considerable dimensions and thus modeling becomes a rather challenging task. By introducing a set of basic instructions, we guide the modeler through a standardized way of reasoning to enhance applicability and guarantee comparability across different models.

Depending on the specific task, we recommend to choose an appropriate perspective of interest. Such perspectives can be a clear representation of the entire platform ecosystem or a detailed description of individual actors, or value propositions. If, for example, the modeler's focus is on providing a high-level overview of the actors in the platform ecosystem, he or she can model from an ecosystem-as-affiliation perspective [1]. As a result, resources, activities and value propositions move out of focus. This may make sense for platforms geared towards a single value proposition where success is largely based on the quantity of actors on each side.

Other platforms, for example, act as marketplaces to distribute numerous applications. Such platforms might benefit from modeling them from an ecosystem-as-structure perspective, focusing on the links between actors and value propositions and the activities and resources they contribute [1]. However, with every application having its own value proposition and relying on specific boundary resources, representing each can get out of hand fairly quickly. The result is an overwhelming model. For some modelers, this degree of detail is not necessary and thus can be omitted without significant loss of information. Others, who strive to visualize all actors and their links on a specific app, should delve in considerably deeper while simultaneously neglecting not involved actors and boundary resources.

As such, PEML can be used to model platform ecosystems on multiple layers. Beginning with a high-level model, the modeler describes all basic components such as actors, sides, boundary resources, and general links that constitute a platform ecosystem. PEML predominantly focuses on this high level of consideration, but also offers components to enable a much more detailed elaboration. Thus, the models can be augmented with additional sub-models, depending on specific purposes. In these sub-layers, the modelers can zoom in on elements of particular interest and take a more detailed perspective as already discussed earlier.

5 Demonstration

5.1 Airbnb

Airbnb is a two-sided platform that enables private persons to offer accommodation and activities to travelers [3]. To create a simple and comprehensible initial

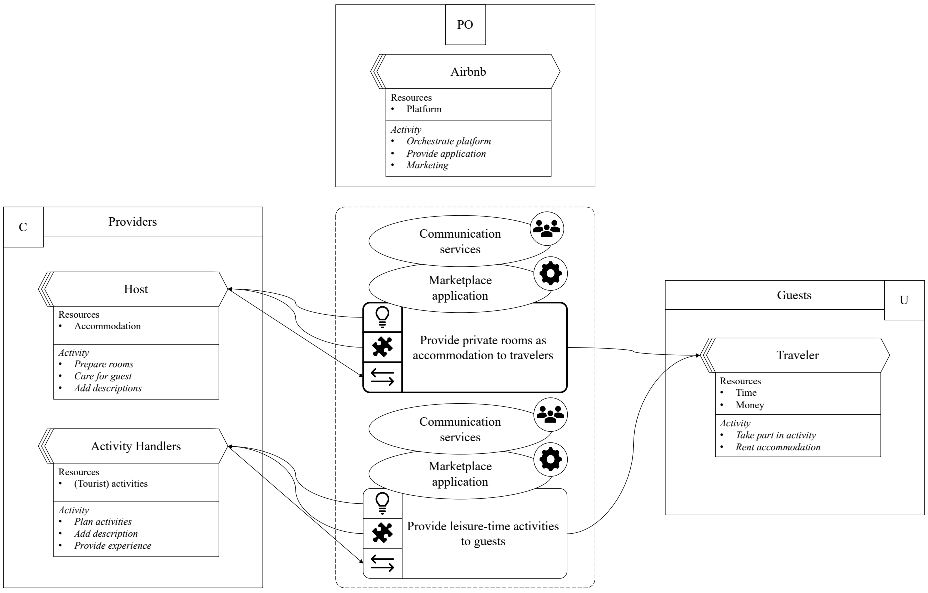

Fig. 4. Demonstration of PEML with Airbnb

example, we only focus on direct actors of the platform. Fig. 4 depicts the model we created for the Airbnb platform.

The first glance at the model shows the two sides of the Airbnb platform. Furthermore, the model unveils two value propositions. Providing accommodation via the platform is the original idea of the platform [4], wherefore we mark it as the main value proposition. Later, Airbnb added the distribution and organization of activities to the platform. By enabling its complementors to sublet their own property to travelers, Airbnb could grow at a great pace and was by 2017 already offering more rooms than the biggest five hotel brands worldwide combined [12]. Additionally, Airbnb also leverages the innovativeness of their ecosystem, resulting in a large variety of accommodations offered, ranging from mansions to tree houses. The same is true for activities, where the number and variety are also high because of the reliance on many complementors. However, the main purpose of Airbnb is to connect hosts with guests and facilitate easy transactions. As evident from the description, both value propositions provide production, innovation and transaction leverage.

The value propositions share the same boundary resources, i. e., the marketplace application as well as the communication services. Airbnb as platform owner provides access to the platform, orchestrates the interaction on the platform, and acquires new users. The users either rent property or attend certain activities. This simplistic demonstration explains the basic idea of PEML to put the platform in the middle of the surrounding actors and only include actors with direct links to the platform.

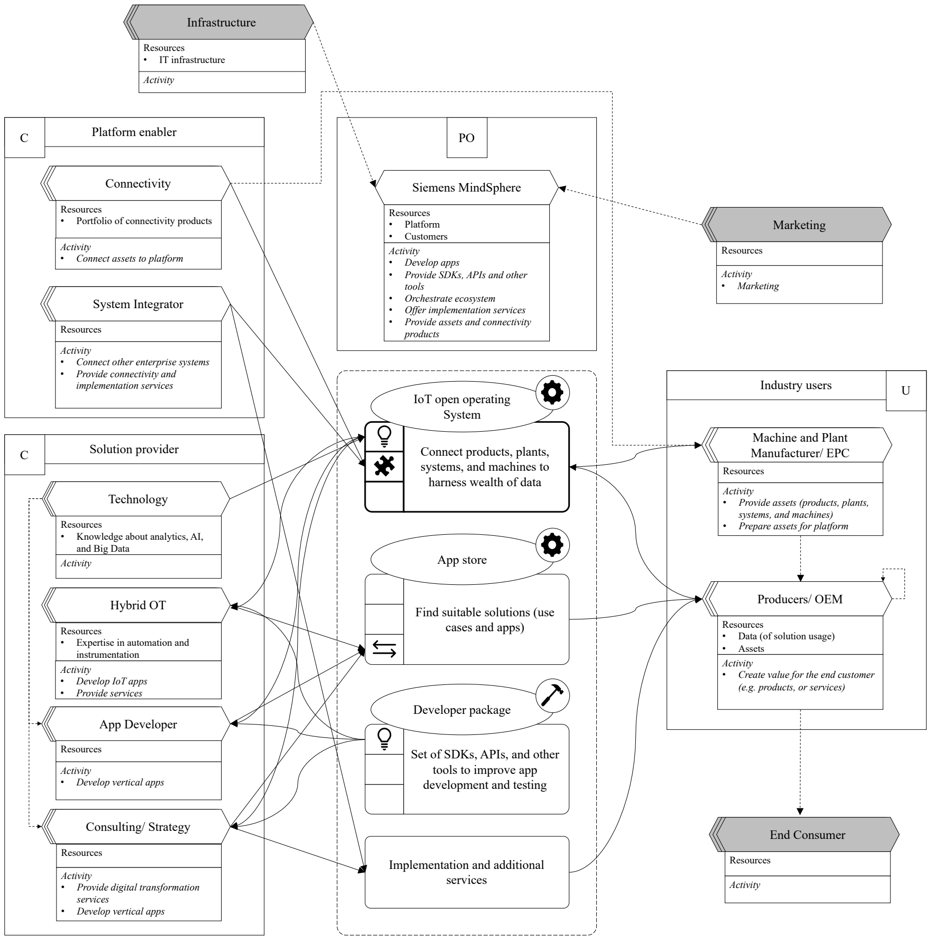

Fig. 5. Demonstration of PEML with Siemens MindSphere

5.2 Siemens MindSphere

Siemens AG is a multinational technology corporation providing products and services for manufacturing. In 2017, Siemens established its platform *MindSphere* to harness the vast amount of data created by their customers [29, 27, 28]. Since then, MindSphere has become a leading platform in the field of Industrial IoT. Figure 5 displays the platform MindSphere using PEML. MindSphere's features allow for the collection of data from various products, plants, systems, and machines. Furthermore, it enables transferring and processing of data either in the cloud or the edge. Siemens provides boundary resources, like application programming interfaces (APIs), and software development kits (SDKs) to leverage the development of custom applications. The MindSphere app store offers third parties to create, test and sell applications, and to access data.

The MindSphere ecosystem bears considerable complexity due to the quantity of complementors and the high variance of their functions and interests [28, 30]. We have identified three basic sides, two of which are complementors. There are two types of customers on the users side: Producers and Machine and Plant Manufacturers, which in turn supply the Producers. The platform enabler side comprises the complementors Connectivity Partners and System Integrator Partners which ensure the technical operability of MindSphere for the customer. Together, Technology Partners, Hybrid OT Partners, App Developers and Consulting/ Strategy Partners form the side of solution providers. They create value by developing apps, implementing use cases or making their domain-specific knowledge available. Additionally, the MindSphere ecosystem also includes actors that do not leverage the platform directly, but are indirectly linked with another actor: Infrastructure Partners, Marketing Partners, and End Consumers.
Of course, this demonstration only shows a part of the overall MindSphere ecosystem. As described in the instructions, the modeler can choose to apply a wider or narrower lens depending on his or her needs. While a narrower lens might enable a more thorough analysis of a small number of value propositions and actors, a wider lens will yield a more high-level overview.

6 Related Works

Various other methods have been proposed or used to model platform ecosystems in a broader sense, especially in the software business [11].

Domain specific languages include, for example, Software Supply Network Diagrams (SSN). As the name suggests, SSN was developed to model networks of organizations that cooperate to provide an integrated software product or service to the market [15, 7]. Moving away from linear supply chains, SSN considers the prevalence of horizontal networks of actors also reflected in ecosystems. However, based on the underlying concept of supply networks, the focus of this modeling approach is on the supply side as opposed to a balanced consideration of all sides of a platform ecosystem.

The majority of languages employed for the modeling of platform ecosystems are not domain-specific. Examples include the i* modeling technique [34] or value network diagrams [5]. The most widely used modeling language is e^{3}value [14]. e^{3}value is very powerful and versatile and can also be used to model platform ecosystems. A similarly comprehensive modeling approach is offered by the value delivery modeling language (VDML) [21]. However, as generic modeling languages, they lack functionality to model some platform-specific features such as sides or boundary resources that are critical for understanding value creation in platform ecosystems, e.g. in terms of network-effects and governance.

Nevertheless, other languages and techniques can and should be used to complement our approach and mitigate some of its shortcomings. For example, e^{3}value can serve to model interactions and especially flows of money and resources within sub-ecosystems around a certain value proposition. Similarly, in case of service offerings involving smart devices and machines (e. g., in the case of Siemens MindSphere), the smart service systems modeling language proposed by Huber et al. [13] might be helpful to capture service-specific aspects.

7 Conclusion

This paper presents the domain-specific modeling language PEML which comprises abstract and concrete syntax, and instructional guidelines to model platform ecosystems. We demonstrate the modeling language using two different platforms to show that it allows for an appropriate specification of the real-world phenomenon across various domains.

This paper entails two contributions. First, we contribute to academia by providing researchers with means to specify their objects of interest when referring to platform ecosystems. In doing so, we answer the recent call for the development of means to "visualise structure [and dynamics] of ecosystems" [24, p. 129]. As a limitation, however, PEML focuses on the static structure and does not necessarily consider platform ecosystem dynamics.

Second, we enable practitioners to model platform ecosystems. This is important in light of the increasing prevalence of platforms in traditionally non-platform domains. As a result, vertical supplier-customer relationships are transformed into horizontal relationships with complementors [23]. Practitioners can use PEML to make sense of this new reality. However, as a caveat, as the focus of PEML lies on digital platforms, its applicability for the modeling of non-digital platforms might be impeded to some extent. Nevertheless, as all core aspects of platforms are represented, it will still be applicable in these cases.

References

1. Adner, R.: Ecosystem as structure. Journal of Management **43**(1), 39–58 (2017). https://doi.org/10.1177/0149206316678451
2. Adner, R., Kapoor, R.: Value creation in innovation ecosystems: how the structure of technological interdependence affects firm performance in new technology generations. Strategic Management Journal **31**(3), 306–333 (2010). https://doi.org/10.1002/smj.821
3. Airbnb: About us - airbnb newsroom (2020), https://news.airbnb.com/about-us/, last visited 29.05.2020
4. Airbnb: Fast facts - airbnb newsroom (2020), https://news.airbnb.com/fast-facts/, last visited 29.05.2020
5. Allee, V.: Value network analysis and value conversion of tangible and intangible assets. Journal of Intellectual Capital **9**(1), 5–24 (2008). https://doi.org/10.1108/14691930810845777
6. Bianco, V.D., Myllärniemi, V., Komssi, M., Raatikainen, M.: The role of platform boundary resources isoftware ecosystems: A case study. In: 2014 IEEE/IFIP Conference on Software Architecture. pp. 11–20 (2014)
7. Boucharas, V., Jansen, S., Brinkkemper, S.: Formalizing software ecosystem modeling. In: Di Cosmo, R., Inverardi, P. (eds.) Proceedings of the 1st international

workshop on Open component ecosystems - IWOCE '09. p. 41. ACM Press, New York, New York, USA (2009). https://doi.org/10.1145/1595800.1595807
8. Frank, U.: Domain-specific modeling languages: Requirements analysis and design guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., Bettin, J. (eds.) Domain Engineering, pp. 133–157. Springer Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-36654-3_6
9. Gawer, A., Cusumano, M.A.: Industry platforms and ecosystem innovation. Journal of Product Innovation Management **31**(3), 417–433 (2014). https://doi.org/10.1111/jpim.12105
10. Ghazawneh, A., Henfridsson, O.: Balancing platform control and external contribution in third-party development: the boundary resources model. Information Systems Journal **23**(2), 173–192 (2013). https://doi.org/10.1111/j.1365-2575.2012.00406.x
11. H. Sadi, M., Yu, E.: Designing software ecosystems: How can modeling techniques help? In: Gaaloul, K., Schmidt, R., Nurcan, S., Guerreiro, S., Ma, Q. (eds.) Enterprise, Business-Process and Information Systems Modeling. pp. 360–375. Springer International Publishing, Cham (2015)
12. Hartmans, A.: Airbnb now has more listings worldwide than the top five hotel brands combined (2017), https://www.businessinsider.com/airbnb-total-worldwide-listings-2017-8
13. Huber, R.X.R., Püschel, L.C., Röglinger, M.: Capturing smart service systems: Development of a domain–specific modelling language. Information Systems Journal **29**(6), 1207–1255 (2019). https://doi.org/10.1111/isj.12269
14. J, G.: e-business value modelling using the e3-value ontology. In: Currie, W.L. (ed.) Value Creation from E-Business Models, pp. 98 – 127. Butterworth-Heinemann, Oxford (2004). https://doi.org/10.1016/B978-075066140-9/50007-2
15. Jansen, S., Brinkkemper, S., Finkelstein, A.: Providing transparency in the business of software: A modeling technique for software supply networks. In: Camarinha-Matos, L.M., Afsarmanesh, H., Novais, P., Analide, C. (eds.) Establishing the Foundation of Collaborative Networks. pp. 677–686. Springer US, Boston, MA (2007)
16. Kapoor, R.: Ecosystems: broadening the locus of value creation. Journal of Organization Design **7**(1) (2018). https://doi.org/10.1186/s41469-018-0035-4
17. Katz, M.L., Shapiro, C.: Network externalities, competition, and compatibility. The American Economic Review **75**(3), 424–440 (1985)
18. Lusch, R.F., Nambisan, S.: Service innovation: A service-dominant logic perspective. MIS Quarterly **39**(1), 155–175 (2015). https://doi.org/10.25300/MISQ/2015/39.1.07
19. Moody, D.: The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering **35**(6), 756–779 (2009). https://doi.org/10.1109/TSE.2009.67
20. Nordstrom, G., Sztipanovits, J., Karsai, G., Ledeczi, A.: Metamodeling-rapid design and evolution of domain-specific modeling environments. In: Proceedings ECBS'99. IEEE Conference and Workshop on Engineering of Computer-Based Systems. pp. 68–74. IEEE (1999). https://doi.org/10.1109/ECBS.1999.755863
21. Object Management Group: Value Delivery Modeling Language 1.1 (2018), https://www.omg.org/spec/VDML/1.1/PDF, last visited 12.05.2020
22. Osterwalder, A., Pigneur, Y.: Business Model Generation: Ein Handbuch für Visionäre, Spielveränderer und Herausforderer. Campus Verlag (2011)
23. Parker, G.G., van Alstyne, M.W., Choudary, S.P.: Platform revolution: How networked markets are transforming the economy and how to make them work for you. W W Norton, New York (2016)

24. de Reuver, M., Sørensen, C., Basole, R.C.: The digital platform: A research agenda. Journal of Information Technology **33**(2), 124–135 (2018). https://doi.org/10.1057/s41265-016-0033-3
25. Rochet, J.C., Tirole, J.: Platform competition in two-sided markets. Journal of the European Economic Association **1**(4), 990–1029 (2003). https://doi.org/10.1162/154247603322493212
26. Rosemann, M., Green, P.: Developing a meta model for the bunge–wand–weber ontological constructs. Information Systems **27**(2), 75 – 91 (2002). https://doi.org/https://doi.org/10.1016/S0306-4379(01)00048-5
27. Siemens AG: MindSphere: So treibt die Industrie weltweit ihre digitalen Transformationen voran (2018), https://www.plm.automation.siemens.com/media/global/de/Siemens-MindSphere-Overview-DE-wp-73922-A21_tcm53-58983.pdf, last visited 12.05.2020
28. Siemens AG: Pressekonferenz Gründung MindSphere World: Press release No. 4 (2018), https://mindsphereworld.de/pressemeldung-4/, last visited 12.05.2020
29. Siemens AG: MindSphere: The cloud-based, open IoT operating system (2020), https://siemens.mindsphere.io/en, last visited 12.05.2020
30. Smith, D.: Mindsphere partner program introduction (2019), https://siemens.mindsphere.io/content/dam/mindsphere/partners/program/MindSphere_Partner_Program_Introduction_2019.pdf, last visited 25.05.2020
31. Thomas, L.D.W., Autio, E., Gann, D.M.: Architectural leverage: Putting platforms in context. Academy of Management Perspectives **28**(2), 198–219 (2014). https://doi.org/10.5465/amp.2011.0105
32. Tilson, D., Sorensen, C., Lyytinen, K.: Platform complexity: Lessons from the music industry. In: 46th Hawaii International Conference on System Sciences. pp. 4625–4634. IEEE (2013). https://doi.org/10.1109/HICSS.2013.449
33. Trabucchi, D., Buganza, T.: Fostering digital platform innovation: From two to multi–sided platforms. Creativity and Innovation Management (2019). https://doi.org/10.1111/caim.12320
34. Yu, E.S.K., Deng, S.: Understanding software ecosystems: A strategic modeling approach. In: Proceedings of the Third International Workshop on Software Ecosystems, Brussels, Belgium, June 7th, 2011. pp. 65–76 (2011)