Ecosystem Business Models and Architectures

John D. McGregor[1], **Simone da Silva Amorim**[2]

[1]School of Computing
Clemson University
Clemson, SC 29634 USA

johnmc@clemson.edu

[2]Federal Institute of Education,
Science and Technology of Bahia
Salvador, BA Brazil 40170

simone.amorim@ifba.edu.br

Abstract. *The ecosystem strategy has provided organizations with time and cost savings by facilitating collaboration with other organizations with similar product ideas and compatible business models. This strategy requires a compatible software architecture that is extensible, flexible, and scalable. In this position paper we clarify definitions, summarize our previous work, and describe our ongoing work that supports defining effective and efficient ecosystem architectures and business models.*

1. Introduction

Every organization that produces a software product is embedded in an ecosystem that comprises its collaborators, competitors, suppliers, and customers. We use "ecosystem" as an analogy to the natural environment in which predators and prey interact. The ecosystems in which we are interested are socio-technical ecosystems in which organizations, people, and technologies collaborate and compete on the development of software-intensive products. The ecosystem may simply be the natural customer-supplier business interactions of one organization with those with whom it does business or it may be a carefully orchestrated strategy that involves legal entities such as open source foundations or partner programs, formal charters, and fees for membership.

Each organization in the ecosystem is following a business model that is more collaborative, more continuous, more personalized, and more transparent than traditional counterparts [Prahalad and Krishnan 2008]. Businesses are involving customers in the product design process much more deeply than a few focus groups. By using social media the opinions, complaints, and suggestions of customers are available to product designers as are their purchasing patterns. Rather than waiting for customers to initiate another purchase, organizations establish a continual dialog via newsletters, reminders of events, and other occasional communication. The data gathered from these continuous, collaborative relationships support the design of products that can be personalized based on data collected in the ecosystem. The design process becomes more transparent to customers as more information is revealed on both sides to establish the continuous relationship.

The software architecture underlying an ecosystem structure is critical to long term success because it unifies the work of personnel who are distributed over different organizations and over geography. In our previous work we have illustrated the need for the architecture of products, built following an ecosystem-based strategy, to be *flexible*, *extensible*, and *scalable* [da Silva Amorim et al. 2013] [McGregor et al. 2013] [da Silva Amorim et al. 2014a] [da Silva Amorim et al. 2014b]. The products must be sufficiently flexible to allow each ecosystem member to configure easily the core product to meet the needs of their particular customers. The architecture must be sufficiently extensible to allow completely new features to be added to the core with minimal modifications. The architecture should be sufficiently scalable to support the anticipated growth over the product life cycle.

We have observed that productive and robust ecosystems are centered around explicitly defined business models and a software architecture designed to satisfy those business models. There needs to be explicit linkages between the characteristics of the ecosystem strategy, business models, and the quality attributes of the product architectures.

The contribution of this paper is an analysis of the interactions of the three quality attributes, flexibility, extensibility, and scalability, with popular ecosystem business models. We discuss the mechanisms that support these attributes and the tradeoffs that result in products with these attributes as high priorities. We illustrate with examples of specific architectures. The remainder of this paper is structured as follows: section 2 presents background relevant to understanding the analysis, section 3 presents an analysis of quality attribute interactions, and section 4 summarizes our work and describes next steps.

2. Background

2.1. Business models

A successful business model has a customer value proposition, profit formula, key resources, and key processes [Johnson et al. 2008]. We will use these elements as we analyze the impact of architecture qualities on business models in the next section. We briefly present three business models, which will be used in the analysis, and we will provide more details during the analysis discussion.

2.1.1. Coopetition

The "coopetition" business model is an efficient product development strategy that leverages the collaboration among organizations in the ecosystem to share effort and risks among the group. The group of organizations identify a core architecture that they wish to share among themselves, sometimes referred to as a platform [Baldwin and Woodard 2008]. The group collaborates in the planning and design of the core architecture. They implement the core and often add frameworks that they think will be useful in building products. These same collaborators then individually compete with each other by taking the commonly developed artifacts and extending or configuring them to produce a product of interest to their specific customer base. The coopetition model's value proposition addresses faster time to market and shared risk; its profit formula is based on software reuse; key resources such as intellectual property (IP) surrounding the core architecture which must be managed and key processes such as management by consensus and promotion by meritocracy. For example, BMW applies the coopetition business model to developing automotive systems through the AUTOSAR Partnership, which has over 100 member organizations. The basic artifact is an automotive platform from which other vehicles can be built. BMW benefits because it can share some of the risk of a new model with competitors such as Ford and GM and because the extensive reference architecture allows BMW to produce more quickly and cheaply than rivals that are not in the partnership.

2.1.2. Multi-sided markets

Hagiu and Wright defines a multi-sided market as "an organization that creates value primarily by enabling direct interactions between two (or more) distinct types of affiliated customers" [Hagiu and Wright 2011]. In this model, an organization creates a product that brings together two other groups that directly interact. Most often the product developer also establishes the market that creates value by having two types of clients. The clients also provide feedback on improving the software supporting the market. This business model's value proposition is to arrange for information suppliers and consumers to be able to find each other; the profit formula is either fee-based or advertising-based; the key resources are the software platform and members of the two client pools; the key processes are developing the software and attracting clients. For example, the Science Exchange provides a web portal where laboratories can register services which they offer [Science Exchange 2014]. Research groups can use the portal to locate services that they may need as part of their experimentation.

2.1.3. Partner program

A large organization may establish a "partner" program whose membership is open to other organizations interested in collaboration on one or more products. These organizations often pay a fee in order to gain the advantage of early access to new releases of a product or special access to source code or other artifacts which they can use to build extensions and apps. This early access allows the partners to have new products synchronized with the release of a new version of the main product; however, partners usually do not have special access to source code and use the core application programming interfaces (APIs) to build products. The value proposition is being first to market, which is an important advantage [Popp 2011]; the profit formula can be fee-based, in-kind contributions, or based on the product extensions attracting more buyers to the core product; the key resources are the plans and designs of the dominant organization; key processes are attracting partners and information sharing processes. For example, Microsoft has a partner program for many of its products [Microsoft 2014]. Although definitely not open source, Microsoft teams gather input from a variety of sources including the partners and build roadmaps of products and features. Partners' early access does not give them any special access to source code; however, they are given access to the planned APIs earlier than non-partners.

2.2. Quality Attributes

Based on our experience with a number of ecosystems in a number of domains, we identified three quality attributes of the core architecture that contribute to the success of an ecosystem: Extensibility, Scalability and Flexibility. We have investigated how these quality attributes are present in ecosystem architectures. [da Silva Amorim et al. 2013][da Silva Amorim et al. 2014a][da Silva Amorim et al. 2014b] present the findings for each attribute and describe their impact in ecosystem architectures.

In our first study about Extensibility[da Silva Amorim et al. 2013] we investigated the APIs of ecosystem platforms. APIs are the common mechanism that is used to allow that external developers to connect their applications to the platform. A well-designed API is essential to keeping external developers productive and satisfied. We identified 4 features with considerable influence in the building of an efficient API: Completeness, Complexity, Design and Documentation. A complete API contains all the API elements necessary to make developers' work easier; the complexity of an API is reduced by eliminating details not needed to understand how to use the functions; the design of the API influences how the components are organized and how decisions are made during building process; the documentation of the API offers guidance for external developers if it is well-organized and clear. We examined these APIs features in 3 ecosystem platforms, Hadoop [Monteith et al. 2013] that is increasing in use, Eclipse [Wermelinger 2009] that is in solid use and CORBA [Schmidt 1993] that is declining in use. We evaluate how these features are present in these architectures in order to illustrate extensibility in these ecosystem architectures.

In our second study[da Silva Amorim et al. 2014a] we investigated the characteristics that contribute to build a scalable ecosystem. We used the six features defined by Bondi [Bondi 2000] that describe the types of scalability in general systems: load, space, space-time, structural, distance and speed/distance. We extended this classification and defined more two additional features: artifact and people. Artifact scalability concerns the increased number of managed artifacts over time. This increase occurs over chronological time, when several organizations join to ecosystem, over and over the phases of the development process as they add new features to the platform. People scalability relates to the ecosystem's ability to manage the growth of the membership of people and organizations. These new participants contribute to diversity in the environment and improve the robustness and sustainability of the ecosystem. In addition, we performed case studies with two ecosystems, the Eclipse Foundation [Wermelinger 2009] and Apache OODT [Apache Foundation 2014] to examine these features in real-world ecosystems.

In our latest study [da Silva Amorim et al. 2014b], we analyzed features that influence the flexibility of ecosystem architectures. These features were classified into two dimensions: business and technical dimensions. The business dimension refers to interactions among internal and external developers in addition to political and economical aspects that influence design decisions that allow the building of flexible architectures. The technical dimension considers the technical attributes with special attention to the explicitly defined features of the architecture. We used Baldwin's approach that is based on the Propagation Cost metric [Baldwin et al. 2013] to adapt for ecosystems architectures. The original approach uses the design structure matrix (DSM) to record dependencies among the modules of system and to calculate the propagation cost considering this cost as a measure of the system flexibility [MacCormack et al. 2006]. It does not consider the dependencies of the API connected to the platform. We added weights in-side the matrix where the values represent dependence with a module of the API. This is the first step to get a different metric specific for ecosystem architectures. We applied this metric in the Apache OODT ecosystem to exemplify the adaptation of the flexibility [Apache Foundation 2014].

3. Analysis

An ecosystem is successful if it continues to attract participants and is compatible with their business models. This motivates the need for a platform that can meet the needs of a wide range of participants. The platform is based on a core architecture that is scalable, flexible, and extensible. Given the basics defined in section 2 we now consider how these qualities support three ecosystem business models.

3.1. Business model characteristics

Table 1 provides a brief description of the linkages between the business process characteristics described in [Prahalad and Krishnan 2008] and the three business models we are investigating.

Table 1. Business models vs. properties

	Coopetition	Multi-sided Market	Partners
Collaborative	Ecosystem members co-develop products and compete on the deployment of those products.	An ecosystem member collaborates with two consumer markets to bring them together.	The dominant producer collaborates with smaller producers.
Continuous	Feedback on innovations from product users informs platform developers.	Feedback from both markets informs the platform supplier.	Ecosystem members work to attract new clients, maybe even individuals, into the product innovation process.
Personalized	Platform may be used directly by individuals instead of other product builders.	Participants can filter a market to receive only the information they wish to have.	Users select the partners they wish to work with and the add-ons they want to use.
Transparent	Open source development ecosystems show everything from rules to voting.	Links to outside information energize both markets through competition.	Published categories of membership instead of individual transactions level the playing field.

3.2. Analysis by business model feature

The three quality attributes address each of the four parts of a business model but in different ways for different models.

3.2.1. Value Proposition

The basic value proposition for ecosystem strategies includes reduced costs, quicker delivery of new features, and improved quality. These benefits are realized from improved synergies among members of the ecosystem and are realized by both developers and users.

Flexible - The coopetition business model and the multi-sided markets model both benefit from the flexibility provided by the variation mechanisms in the core architecture to reconfigure the core to address multiple audiences. Most partner programs provide less flexibility and the partners only build extensions rather than reconfiguring the core product.

Extensible - The coopetition business model has less need for extensibility because the intense interactions in designing the core platform gives opportunities for anticipated changes to be identified and accommodated via variation mechanisms. The multi-sided markets business model uses extensibility when a specific client population can be attracted by domain-specific features such as background checks for a dating service.

The partner programs extend the core product but market timing is controlled by the core product.

Scalable - The multi-sided markets and partner models are based on handling specific client loads so scalability is critical. The coopetition model is more generic and certain types of products might be load sensitive such as the size of the product to be built in a tools ecosystem.

3.2.2. Profit formula

The profit formulae for ecosystem strategies vary across strategies and roles of organizations, but what is common is that most strategies use an indirect formula. That is, the organizations using many of the ecosystem strategies do not directly receive payment for some of the work that is done. The work on the core platform is, in most cases, pooled for the common good, often as open source. The profit may come from delivering an audience for advertising-based models or from the ability for each participant to use the work of the whole group as the basis for future products.

Flexible - The flexibility of the core platform enhances the profit formula of an organization by giving the organization options for future products and helps attract and retain developers. The effort required for inserting the variation mechanisms and for later exercising these options is a cost against that profit. The coopetition model is particularly sensitive to how flexible the platform is since the expectation is to produce products as rapidly as possible.

Extensible - The extensibility of the core platform is the key to profit in the ecosystem architectures. Anticipating all future features is impossible; however, ecosystem architects manage future feature development through the insertion of variation mechanisms. The use of loose coupling and well-defined APIs reduce the cost of defining the extensions required for products and thereby increase profits.

Scalable - Products that utilize the multi-sided markets strategy will need to scale in multiple dimensions in order to profit. Being able to handle an unconstrained number of participants on each side of the market will yield the best profit.

3.2.3. Key resources

Personnel, money, and facilities are the key resources specified in most ecosystem business models. The facilities in particular include tools for collaboration and social networking tools. The ecosystem is as strong as the interactions among the organizations and between the organizations and its customers.

Flexible - Developers who were not involved in building the core can become productive more rapidly due to the variation mechanisms in place in the flexible architecture. This conserves all of the key resources. The multi-sided markets strategy follows a distinct pattern which can be reflected in variation mechanisms and recipes to speed development and conserve resources.

Extensible - Well-defined APIs in an extensible architecture make it easy for organizations following a partner strategy to attach their product extension to the main product. Up-to-date documentation, example architecture instantiations, and other aids assist the developer in building product extensions.

Scalable - The multi-sided market strategy is sensitive to scaling from a resource perspective. The searching and matching that usually is the focus of such a strategy needs an architecture that maintains a linear growth pattern.

3.2.4. Key processes

The ecosystem business models usually include collaborations such as peer-to-peer, dominant-to-subordinate, and multi-way interactions. The models also include development processes for both the core platform and products that are extensions of the core.

Flexible - The explicit variation mechanisms inserted in the core product to make it flexible allows the development of the products to follow a set pattern making the process easy to define and follow. Ecosystems using the coopetition strategy select the variation mechanisms that first fit the situation and then that are the easiest to exercise.

Extensible - The partner strategy will benefit the most from having a standard set of processes for defining and then for using the APIs in the architecture because partners are usually limited to accessing the product through APIs to define extensions. In the coopetition model the organization who developed the core is also developing the product and should have some residual knowledge that goes beyond the APIs and should have sufficient access to benefit from it.

Scalable - The development processes must accommodate a changing number of organizations and their contributed resources. The processes must be able to scale in a distributed environment. For the ecosystem to be successful the development process must be able to support increases in number of organizations, number of developers, and number of projects.

4. Summary

The ecosystem strategy supports a number of unique business models and enhances some traditional ones as well. Each model supports different value propositions, profit is realized in different ways, and key resources and processes impose different constraints. The architects and developers building the core architecture need to understand the business model that ultimately will be the guide to success in order to prioritize how to handle variation points, API definitions, and algorithm growth rates.

Our future work will include defining measures which distinguish ecosystems that possess differing degrees of flexibility, extensibility, and scalability and does so in units that are useful to ecosystem participants. The processes will have to address the wide variation in how the core architecture will ultimately be used. We will also identify ecosystem design patterns.

References

Apache Foundation (2014). Apache oodt. `http://oodt.apache.org/`.

Baldwin, C., MacCormack, A., and Rusnak, J. (2013). Hidden structure: Using network methods to map product architecture. Technical Report Working Paper 13-093, Harvard Business School.

Baldwin, C. Y. and Woodard, C. J. (2008). The architecture of platforms: A unified view. Technical Report 09-034, Harvard Business School.

Bondi, A. B. (2000). Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd international workshop on Software and performance*, pages 195–203.

da Silva Amorim, S., Almeida, E. S. D., and McGregor, J. D. (2013). Extensibility in ecosystem architectures: an initial study. In *WEA*, pages 11–15.

da Silva Amorim, S., de Almeida, E. S., and McGregor, J. D. (2014a). Scalability of ecosystem architectures. In *WICSA*, pages 49–52.

da Silva Amorim, S., McGregor, J. D., Almeida, E. S. D., and von Flach G. Chavez, C. (2014b). Flexibility in ecosystem architectures. In *WEA*.

Hagiu, A. and Wright, J. (2011). Multi-sided platforms. Technical Report 12-024, Harvard Business School.

Johnson, M. W., Christensen, C. M., and Kagermann, H. (2008). Reinventing your business model. *Harvard business review*, 86(12):57–68.

MacCormack, A., Rusnak, J., and Baldwin, C. Y. (2006). Exploring the structure of complex software designs: An empirical study of open source and proprietary code. *Management Science*, 52(7):1015–1030.

McGregor, J. D., Monteith, J. Y., Amorim, S., and Almeida, E. (2013). Modeling the contributions of software architecture to the success of an ecosystem. In *Proceedings of SATURN - SEI Architecture Technology User Network*.

Microsoft (2014). Microsoft partner program. `https://mspartner.microsoft.com/en/us/Pages/index.aspx`.

Monteith, J. Y., McGregor, J. D., and Ingram, J. E. (2013). Hadoop and its evolving ecosystem. In *Proceedings of the Fifth International Workshop on Software Ecosystems*, pages 57–68.

Popp, K. M. (2011). Hybrid revenue models of software companies and their relationship to hybrid business models. In *IWSECO@ICSOB*, pages 77–88.

Prahalad, C. and Krishnan, S. (2008). *The New Age of Innovation: Driving Cocreated Value Through Global Networks*. McGraw Hill professional. McGraw-Hill Education.

Schmidt, D. C. (1993). The adaptive communication environment: An object-oriented network programming toolkit for developing communication software.

Science Exchange (2014). `https://www.scienceexchange.com/`.

Wermelinger, M. (2009). The architecture evolution of eclipse. `http://michel.wermelinger.ws/chezmichel/2009/10/the-architectural-evolution-of-eclipse/`.