

# Designing Software Ecosystems: How to Develop Sustainable Collaborations?

## Scenarios from Apple iOS and Google Android

Mahsa H. Sadi<sup>1</sup>(✉), Jiaying Dai<sup>2</sup>, and Eric Yu<sup>1,2</sup>

<sup>1</sup> Department of Computer Science, University of Toronto, Toronto, Canada  
{mhsadi, eric}@cs.toronto.edu

<sup>2</sup> Faculty of Information, University of Toronto, Toronto, Canada

**Abstract.** It has become an increasingly common practice that software companies collaborate with external developers to develop a software platform for a shared market, constituting software ecosystems. One main concern in adopting the practice of software ecosystem is how to attract external developers to a platform, and how to establish sustainable collaborative relationships with them. We discuss that explicating and in-depth analysis of developers' objectives and decision criteria can facilitate the design of sustainable collaborations in software ecosystems. Scenarios from Apple iOS and Google Android ecosystems are used for illustration.

**Keywords:** Software ecosystems · Analysis · Google Android · Apple iOS · Mobile operating system ecosystem · Design · Modeling · Decision making

## 1 Introduction

It has become an increasingly common practice for software development companies to collaborate with external developers in order to develop a software platform for a shared market, founding software ecosystems [1, 2]. In a software ecosystem, a key software platform developer (referred to as the keystone) engages external developers in the promotion of its software platform in various ways such as developing or extending the platform, or providing complementary applications and services for the platform [3].

Various forces lead to the adoption of software ecosystem practice among software companies, including sharing the cost of production, dealing with the diverse and numerous demands of end users, using the domain expertise of external stakeholders to develop domain-specific applications, co-innovating, and fighting against competitors. Moreover, the success of software platforms heavily depends on the number of supporting applications and services [4, 5]. To benefit from the above advantages, it is crucial for a keystone software company to expand its collaboration network in software development and service provision, and to attract and retain as many external developers and collaborators as possible.

The pivotal role that collaboration plays in the success of modern-day software development and provision demands concentrated effort to support the elaborate design and configuration of healthy and sustainable relationships in software ecosystems. Establishing collaborative relationships is a multifaceted problem for a keystone platform developer, spanning various technical, business, organizational, and social concerns that must be addressed simultaneously. A successful software ecosystem needs to have a viable business model, a well-organized inter-organizational interaction model, a well-designed collaborative software development process, and a software platform that enables the collaboration [3, 6–8].

However, despite the widespread adoption of the Software Ecosystem practice among software companies, there is still no rigorous systematic approach for establishing sustainable collaborative relationships in a software ecosystem. Current approaches (e.g. [3, 9]) mostly propose several general rules of thumb to develop software ecosystems. It has been neglected that each software platform and keystone software company has a specific set of characteristics which attracts external developers, and with each software ecosystem specific groups of developers collaborate pursuing a diverse set of business, social, and technical objectives [10]. Moreover, little attention has been given to the systematic structuring and design of collaborations in software ecosystems.

To address the above issue, in this paper, we take two steps:

1. We discuss that in order to create a healthy and sustainable software ecosystem, a keystone platform developer needs to elicit and analyze the objectives and decision criteria of both its organization and external developers for participation.
2. Using real-world scenarios from the smartphone software ecosystem (extracted from [10]), we illustrate how to use the objectives and decision criteria of the collaborators in configuring a sustainable software ecosystem. To model and analyze the objectives of collaborators, we use the *i\** goal-oriented social modeling technique [11].

## 2 Motivating Scenario: The Smartphone Software Ecosystem

To make our discussion concrete, we use common scenarios from the smartphone software ecosystem. The scenarios are extracted from a recent study [10] investigating what motivates application developers to join mobile software ecosystems such as Apple iOS and Google Android.

**Scenario.** In the smartphone software ecosystem, the key platform is mobile operating system, mainly developed by keystone software companies such as Google, Apple, or Microsoft. Complementary software applications and services play a crucial role in the success of a mobile operating system in the market. Hence, it is essential for mobile platform developers to establish a large network of external collaborators. External collaborators are mostly engaged in the development of complementary application and software services for the operating system. For this purpose, the keystone software companies usually provide a software development toolkit (SDK) for the platform.

This toolkit provides necessary infrastructure for the external developers to develop software applications [10].

**Question.** One main concern for a keystone mobile platform developer is “How to attract and retain a large network of application developers?” or “How to establish sustainable collaborative relationships with application developers?”

### 3 The Proposed Method

**Generic Approach.** Intuitively, to answer the above question, the mobile platform developer needs to do the following:

1. *Identify different types of application developers that collaborate with the mobile platform developer, and categorize them according to their behavior by understanding their motivations, expectations, and criteria for deciding to join an ecosystem.* Application developers may contribute to a mobile platform for various reasons. These objectives can be technical (i.e. related to the features and quality attributes of the software platform) or non-technical (related to personal, social, or business motivations). For example, some developers join a software ecosystem based on their business motives of selling applications to the end-users of the mobile platform to achieve financial gain, while others join to improve their programming or technical skills. Knowing these expectations and objectives enables the mobile platform developer to seek for alternatives to fulfill them. Moreover, obtaining information about the objectives of application developers and analyzing these objectives avoids premature commitment to some rules of thumb for attracting external developers in a software ecosystem. Clarifying the objectives and decision criteria of developers can be used as a source of information to derive the specific requirements that adequately fulfill the expectations of each group of application developers.
2. *Explicate and analyze the technical and non-technical requirements for designing a sustainable collaboration.* Having gained overall insight into the main groups of external collaborators and their motivations and expectations, the next step will be to refine and analyze the obtained information about each group of application developers as a source for deriving the requirements of an appropriate collaborative environment.
3. *Derive alternative solutions for designing an appropriate collaborative environment that fulfills the elicited requirements.* After identifying the specific requirements of the software ecosystem, the final step is to decide what courses of action should be taken to design or improve the configuration of the collaborations among the keystone software company and the application developers.

**Modeling and Analysis Guidelines.** Although the above steps outline the main activities that address the mobile platform developer’s concerns, they do not provide specific guidance on how these activities can be performed. To perform the above steps, we adopt a model-based approach. We develop a set of guidelines for how to model and analyze application developers’ objectives and criteria in order to establish

sustainable collaborative relationships with them. The following steps identify what should be modeled and what analyses are needed to be performed on the models.

1. *Explicate and model the objectives and decision criteria of the developers who contribute to the software platform.* In this model, the developers, the activities and tasks that these developers perform in the software ecosystem, the factors that influence their decision for choosing a specific platform, and their relationships with the keystone software company should be identified and explicated. This information can be obtained through various ways such as surveying (as used in [10]) or interviewing application developers.
2. *Clarify and refine the objectives and decision criteria to the extent that they can be translated into design requirements for a software ecosystem.* To find concrete design solutions, the objectives and decision criteria of developers need to be refined to the extent that they can be translated into alternative design solutions.
3. *Investigate the degree of the fulfillment of the objectives and decision criteria.* From developers' perspective, their expectations and objectives may be fulfilled to various degrees in a software ecosystem. Therefore, it is required to identify and evaluate the fulfillment of the objectives and decision criteria through gathering data about the perception of the developers from the software ecosystem.
4. *Identify the importance of the objectives and decision criteria and prioritize them.* Having gained insight into the requirements of the software ecosystem, the next step is to identify the importance of these requirements and prioritize them in the software ecosystem. The elicited requirements in previous step may be of different degree of importance and influence both from the perspective of keystone platform developer and the external software developers. Therefore, before coming up with alternative design solutions, the priority and influence of the requirements should be explicated. This type of information can be elicited through various sources or approaches such as interviewing application developers.

## 4 Analyzing and Designing the Smartphone Software Ecosystem

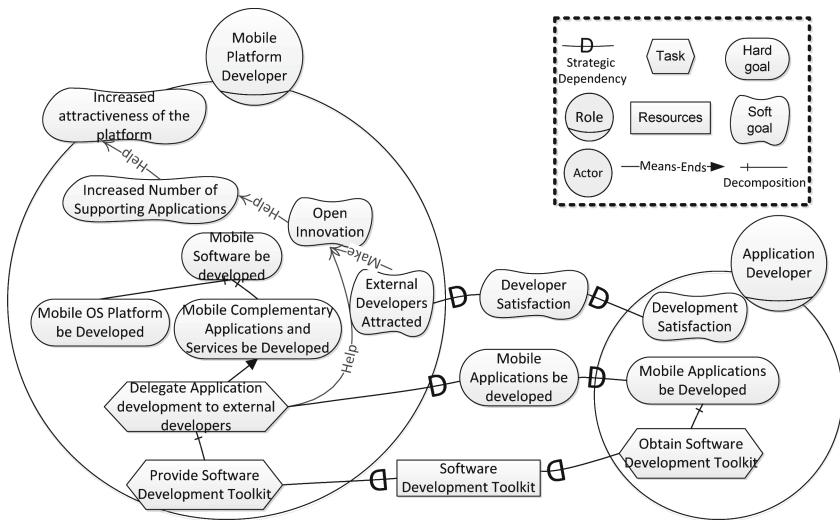
Now, we return to the motivating scenario. We aim to help a mobile platform developer to come up with alternative solutions for structuring sustainable collaborative relationships with application developers, following the proposed method.

In the following, we first explain how to model collaborators' objectives and decision criteria using the  $i^*$  goal-oriented social modeling technique [11] (Sect. 4.1). Then, we apply the proposed method in Sect. 3 to analyze the objectives and decision criteria of collaborators in the smartphone software ecosystem in order to find out appropriate solutions for developing collaborative relationships. Since we need information about the objectives and decisions of collaborators, in Sects. 4.1 and 4.2, we use available scenarios from the Apple iOS and Google Android ecosystems.

### 4.1 Guidelines for Model Construction Using the i\* Technique

In Fig. 1, based on the content of the scenario described in Sect. 2, we have developed a generic model of collaboration between mobile platform developer and application developer in the smartphone software ecosystem. Figure 1 explicates two main collaborators in the mobile operating system ecosystem, the activities and operations of each collaborator, the objectives and motivations of the collaborators, and the dependencies between them. In the following, we explain how this model is developed from the content of scenario:

(a) *Collaborators (modeled as “Roles” or “Actors”)*: In Fig. 1, “Mobile Platform Developer” and “Application Developer” are modeled as Roles. Each role can be occupied by specific mobile platform developers such as Google and Apple or the specific application developers who collaborate with each of these two companies.



**Fig. 1.** Generic model of collaboration between keystone software company and software application developers in mobile operating system ecosystem

(b) *Operations and Activities (modeled as “Goals” and “Tasks”)*: The main operation of “Mobile Platform Developer” is to develop the mobile software (expressed in the goal of “Mobile Software be Developed”). Modeling this operation as goal means that it can be achieved via various alternative activities and tasks. Developing mobile software include two finer-grained activities of developing mobile operating system platform and developing complementary applications and services. In the mobile software ecosystem, the common strategy is to delegate the development of complementary applications to external developers. This approach is demonstrated through “Means-Ends” relationship between the hard goal of “Mobile Complementary

Applications and Services be Developed” and the task of “Delegate Application Development to External Developers”. On the other side, the main activity of “Application Developer” is to develop complementary mobile applications, which requires to obtain the software development kit for the platform. These activities are depicted in the decomposition relationship between “Mobile Application be Developed” goal and “Obtain software development kit”.

(c) *Objectives and Decision Criteria of Collaborators (modeled as soft-goals)*: Capturing motivations and initiatives in soft goals conveys that there is no clear-cut criteria for assessing whether these objectives have been achieved or not. In the mobile software ecosystem, collaborators pursue specific objectives for collaboration. The main objective of “Mobile Platform Developer” is to increase the attractiveness of the platform. To achieve this objective, “Mobile platform developer” pursues open innovation initiatives, explicated as “Open Innovation” soft goal. For Open initiatives to be successful, external developers should be attracted to the mobile platform. This relationship is shown as “Make” relationship between “Open Innovation” and “External Developers Attracted” soft goals. On the other side, application developers would continue to collaborate with the keystone software company if their expectations are satisfied. Although feeling satisfied means differently among various developers, it can be considered as a common objective among all the application developers. This objective is captured in terms of “Development Satisfaction” soft goal in “Application Developer” role.

(d) *Reasoning behind the adoption of specific approaches and tasks by each collaborator*: In mobile software ecosystem, collaborators adopt specific activities for specific reasons. For example, the reason for delegating development of complementary applications to external developers is that it helps the open innovation initiative which in turn helps promote the attractiveness of mobile operating system as a high-level business objective. These relationships are shown as “Help” relationship between the related tasks and soft goals.

(e) *The relationships among collaborators (modeled as “strategic dependencies”)*: Delegating the development of complementary applications to application developers creates specific relationships between the collaborators: “Mobile Platform Developer” depends on the “Application Developer” for the goal of “Complementary applications be developed” to be achieved. On the other side, “Mobile Platform Developer” depends on “Application Developer” for his/her satisfaction from development.

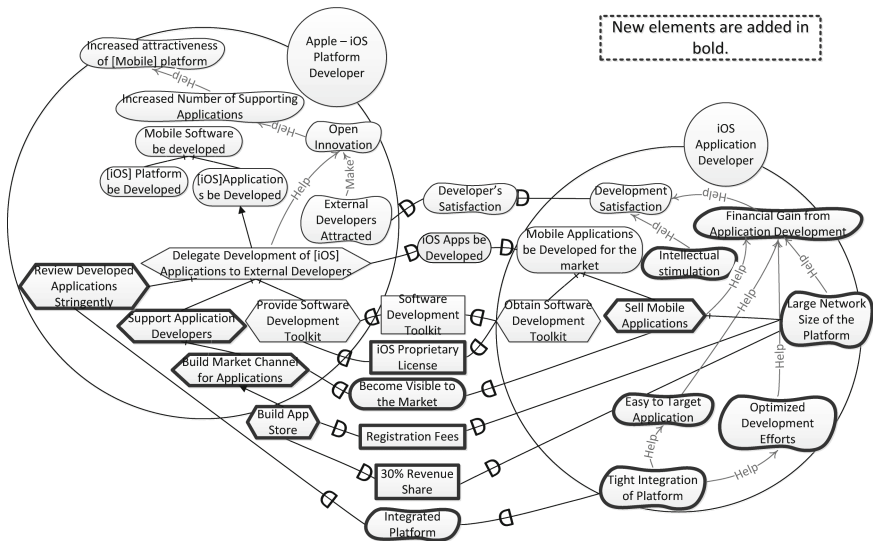
As Fig. 1 illustrates, for the mobile operating ecosystem to be successful, a “Mobile Platform Developer” needs to fulfill “Development satisfaction” in the “Application Developer”. Hence, the mobile platform developer needs to know the different groups of application developers who collaborate with it, understand why they collaborate, and create an environment that motivates them to collaborate.

To explicate and analyze developers’ objectives and motivations, in the following, we build upon scenarios from the mobile platform developer of Apple iOS and Google Android the application developers who develop applications for these two platforms.

### 4.2 The Apple iOS Software Ecosystem

**Scenario. Mobile Platform Developer:** iOS is the Apple operating system for mobile phones and tablets, having been developed since 2007. iOS is exclusively used for smartphones and tablets produced by Apple, its license is proprietary and its source code is not publicly available. To develop applications for iOS, external developers must register in the fee-based “iOS Developers Program”, which enables them to download iOS SDK. Third-party applications are extensively reviewed by Apple to check their compliance with the guidelines set in the iPhone SDK agreement, before becoming visible to the end users via App Store. Apple charges developers for 30 % share of the application sale, but no fee is received for free applications [10].

**Application Developers:** According to the study [10], Apple third-party developers are mainly driven by financial gains. Intellectual stimulation is also an important factor for the developers who join Apple iOS ecosystem. These developers often prefer to charge fee for their application being used by Apple iPhone/iPad end users. The main characteristics of the iOS platform that motivate this group to join Apple iOS ecosystem are as follows: (a) Large network size of the platform (composed of the number of users, the market size, and the number of applications), and (b) the tight integration of the platform. A tightly integrated platform makes the complementary application development process easier for developers with strong motivations in financial gains by optimizing development efforts and facilitating the targeting of the applications.



**Fig. 2.** Collaboration model between keystone software company and software application developers in Apple iOS software ecosystem

**Applying the Method.** (1) *Explicating the objectives of the developers.* As conveyed in the scenario, one major group of developers who contribute to iOS platform are driven by business and financial motivations. In the scenario, “Financial gain from application development” and “Intellectual stimulation” hint to the main motivations of iOS developers (explicated as two soft goals contributing to “Development Satisfaction” soft goal in Fig. 2). Efforts to improve the satisfaction of these social and business soft goals would help enhance development satisfactions of iOS developers’. Moreover, “Tight integration of software platform” soft goal elicits one technical characteristic of iOS platform which is of importance to external developers. A tightly integrated platform contribute to “Optimized development effort” and “Easy to target application” decision criteria of iOS developers. Therefore, finding solutions to improve the integration of the software platform would also increase the development satisfaction of iOS third-party developers.

(2) *Deriving design requirements.* (a) *Clarifying developers’ objectives.* “Financial gain from application development”, “Intellectual Stimulation”, and “Tight Integration of Software Platform” hint to some requirements of Apple iOS software ecosystem. However, to reach to specific design solutions, these requirements need to be further refined and elaborated. For this purpose, questions such as the following should be answered: “What factors influence or increase intellectual stimulation in Apple application developers?”, or “To what factors tight integration of platform refer to?” Elaborating on these questions requires information which are not provided in the above scenario. Therefore, more information needs to be gathered about Apple iOS software ecosystem to elaborate the requirements. (b) *Investigating the fulfillment of the elicited requirements.* The objectives and decision criteria of the iOS developers may be fulfilled to various extent. For example, it is possible that “Tight integration of platform” is fulfilled to a good extent, but “Being intellectually stimulated” could be not satisfactory enough. Therefore, before focusing the design attempts, it is required to gather information about the fulfillment of the elicited requirements. (c) *Prioritize the requirements:* We assume that from among the two non-technical requirements of “Intellectual Stimulation” and “Financial gain from application development”, the latter is probably more important for Apple application developers. Hence, it will be more effective to focus the design efforts to improve the satisfaction of this non-technical requirement among iOS developers.

(3) *Concluding the requirements and reaching to design solutions:* Since Apple iOS developers are mainly motivated by financial gain, “Sell Mobile Applications” is one activity that these developers perform in the software ecosystem. For performing this task, developers become dependent on iOS platform developer, for the goal of “Applications become visible to the market place” (see Fig. 2). Explicating these requirements enables the Apple iOS platform developer to find out appropriate solutions to support the application developers. One solution for supporting external developers is to “Build market channels for applications” which is currently realized by the development of App Store.



**Conclusions from Modeling and Analysis.** As identified in the above study, financial gain is one main requirement to sustain a collaborative relationship between iOS application developers and Apple. Therefore, building a market channel for software applications is one appropriate solution to support iOS external developers. We illustrated that this solution can be concluded by elaborate analysis and investigation of the case of third-party developers who collaborate with iOS.

However, depending on the specific situation of a software ecosystem, solutions for supporting external developers differ. To demonstrate this difference, in the next section, we walk through scenarios from Google Android Software Ecosystem.

### 4.3 The Google Android Software Ecosystem

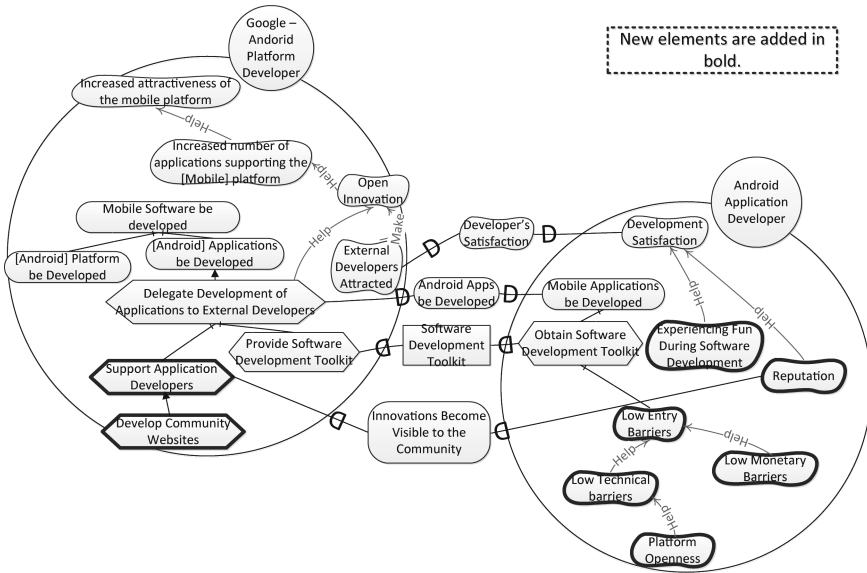
**Scenario. Mobile Platform Developer:** Android OS is developed by Open Handset Alliance – an alliance of 84 companies specialized in software, hardware, and telecommunication led by Google, since 2008 – as an open-source project based on Linux kernel. The open-source strategy aids to increase the adoption of Android OS platform among various mobile device manufactures. A set of software development tool kits are available for Android. External developers can download many of these SDK's for free and without registration. The developed applications undergo a short checking process by Google Employees and are then made available to the market (via Google Play Store). Developers are charged for a non-recurring registration fee to access Google Play. However, there are also other Android app stores besides Google Play store, some of which do not charge developers with fees. The ease of use and high accessibility of free Android SDKs enables Google to attract a wide and diverse range of application developers. Similar to Apple, Google charges the application developers for 30 % of unit sales, but do not charge fees for free applications [10].

*Application Developers:* According to the study [10], low entry barriers (including low monetary and low technical requirements) and platform openness are the major characteristics of Google Android platform that motivates external developers to join this ecosystem. Low monetary barrier refers to the point that Android SDK is free to download. Low technical barrier refers to the point that Android OS is open-source, various SDKs are available for Android, and there is no specific technical requirements or restrictions for using Android SDKs. As a result, one major group of Android application developers are motivated by intrinsic reasons such as improving their programming skills, and having fun during development. Other social reasons, such as gaining reputation in the Android community also motivates this group of developers to choose Google Android ecosystem. Moreover, the majority of Android application developers publish their application for free on Google Play Store.

**Applying the Method.** (1) *Explicating the motivations of developers.* As described in the scenario, one major group who contribute to Google Android platform are driven by intrinsic and social motivations, and they mostly develop Android applications for free. In the scenario, “Experiencing fun during software development”,

gaining “Reputation” among Android developers and end users, and “Low entry barriers” refer to the factors motivating Android application developers.

(2) *Deriving design requirements.* (a) *Clarifying developers’ objectives.* The above explicated objectives need to be further clarified and elaborated to derive the design requirements in Android ecosystem. Similar to the analysis of Apple iOS, questions such as “What factors influence or increase the experience of fun in Android developers”, or “To what factors Low entry barriers refer to?” should be answered. As described in the scenario, “Low entry barriers” consists of sub factors of “Low technical barriers” and “Low monetary barriers”. One factor which contributes to “Low technical barrier” in Android is “Platform Openness”. The refinement of the developers’ objectives is explicated in decomposing soft goals via “Help” links in Fig. 3. (b) *Prioritize the requirements:* From among the explicated requirements, we assume that “Reputation” is more critical than the others. Consequently, attempts to improve the feeling of being recognized in Android developers would contribute to the sustainability of Android software ecosystem.



**Fig. 3.** Collaboration model between Keystone software company and software application developers in Google Android software ecosystem

(3) *Concluding the requirements and reaching to design solutions:* As reasoned about, to sustain collaborative relationships with Android developers, improving the sense of being recognized in the users and developers community (“Reputation” soft goal) is one main requirement. For the “Reputation” soft goal to be satisfied, Android Developers are dependent on Android platform developer” for the goal of “Innovations become visible to the community”. Android platform developer should find out

appropriate solutions to support the application developers. One solution for supporting external developers could be to “Develop Community Websites” to publicize the information about the innovations to the end user and developer community [10].

**Conclusions from Modeling and Analysis.** As identified above, in contrary to Apple iOS, Android application developers choose the open-source platform to cultivate their intrinsic motivations such as skills development and reputation enhancement. Assuming the reputation enhancement as the main requirement for designing sustainable collaborative relationships with Android developers, it was concluded that different approaches should be adopted to support Android developers, such as developing community websites to publicize developers’ innovations.

## 5 Related Research

In this study, building upon our previous attempts [8, 12], we explained how to systematically explicate and analyze the objectives of the members in order to design sustainable and healthy collaborative relationships in software ecosystems. To the best of our knowledge, no similar approach has been proposed in the literature related to software ecosystems. However, there are two main groups of research efforts that are closely related to this study:

*Analysis and Design of Software Ecosystems in General.* Analyzing and designing software ecosystems is a recent and multi-faceted issue. To reduce the complexity of the problems raised in the design of software ecosystems, existing efforts mostly advise to separate business, organizational, social, and technical concerns, and address each of these concerns independently (e.g. [13]). While few efforts (e.g. [13, 14]) analyze a set of these aspects simultaneously but separately, the majority of existing research efforts merely focus on one dimension (e.g. [15] focusing on the technical dimension). Very little attention has been given to (a) analyzing the interrelationships between technical, business, social and organizational factors in the design of software ecosystems; and (b) aligning these dimensions with each other. Herein, we illustrated how to model and analyze the interrelationships among various socio-technical factors in the design of software ecosystems.

*Designing Collaborative Relationships in Software Ecosystems in Particular.* One specific issue in the design of software ecosystems is structuring collaborative relationships with external stakeholders (i.e. other software companies and individual application developers). Only a few systematic approaches and techniques have been proposed to address this issue (e.g. [14, 16, 17]). However, these efforts mainly focus on the business and inter-organizational relationships between the members of a software ecosystem. There are also few studies, (e.g. [18]) that focus on designing technical collaborations among the members. In this study, we illustrated how to model and analyze the overall perception of the collaborators from the socio-technical environment of a software ecosystem to identify appropriate solutions for developing sustainable collaborations.

## 6 Conclusions

One main step in transition into a software ecosystem approach is to attract external software developers to a software platform, and to establish sustainable collaborative relationships with them. Using scenarios from Apple iOS and Google Android software ecosystems, we illustrated how modeling and analyzing developers' objectives help find appropriate solutions for designing sustainable collaborative relationships with external developers.

**Limitations of this study.** (1) This study is performed by post-mortem modeling and analysis of available scenarios on existing software ecosystems. To demonstrate the viability of the developed modeling and analysis guidelines, experimentation in the context of real case studies is required. (2) The developed guidelines outline some analyses that are needed to be performed on collaborators' objectives and decision criteria. These prototype analyses need to be elaborated with specific techniques for eliciting, evaluating, and prioritizing collaborators' objectives and decision criteria.

**Acknowledgements.** Financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Ontario Research Foundation (ORF) are gratefully acknowledged.

## References

1. Jansen, S., Finkelstein, A., Brinkkemper, S.: A sense of community: a research agenda for software ecosystems. In: 31st International Conference on Software Engineering-Companion Volume, ICSE-Companion 2009, pp. 187–190 (2009)
2. Bosch, J.: From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference, Carnegie Mellon University, pp. 111–119 (2009)
3. Jansen, S., Brinkkemper, S., Souer, J., Luinburg, L.: Shades of gray: opening up a software producing organization with the open software enterprise model. *J. Syst. Softw.* **85**(7), 1495–1510 (2012)
4. Iansiti, M., Levien, R.: Strategy as ecology. *Harv. Bus. Rev.* **82**(3), 68–81 (2004)
5. Bosch, J.: Software ecosystems: taking software development beyond the boundaries of the organization. *J. Syst. Softw.* **85**(7), 1453–1454 (2012)
6. Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J., Robinson, W.: The brave new world of design requirements. *Inf. Syst.* **36**(7), 992–1008 (2011)
7. Sadi, M. H., Yu, E.: Designing software ecosystems: how can modeling techniques help? In: Proceedings of 20th Conference on Exploring Modelling Methods for Systems Analysis and Design (EMMSAD), LNBIP **214**, pp. 360–375 (2015). doi:[10.1007/978-3-319-19237-6\\_23](https://doi.org/10.1007/978-3-319-19237-6_23)
8. Sadi, M. H., Yu, E.: Analyzing the evolution of software development: from creative chaos to software ecosystems. In: IEEE 8th International Conference on Research Challenges in Information Science (RCIS), pp. 1–11. IEEE (2014)
9. Popp, K., Meyer, R.: Profit from Software Ecosystems: Business Models Ecosystems and Partnerships in the Software Industry. BoD—Books on Demand, Norderstedt (2010)
10. Koch, S., Kerschbaum, M.: Joining a smartphone ecosystem: application developers' motivations and decision criteria. *Inf. Softw. Technol.* **56**(11), 1423–1435 (2014)

11. Yu, E., Giorgini, P., Maiden, N. (eds.): *Social Modeling for Requirements Engineering*. MIT Press, Cambridge (2011)
12. Yu, E., Deng, S.: Understanding software ecosystems: a strategic modeling approach. In: *Proceedings of 3rd IWSECO*, pp. 65–76 (2011)
13. Christensen, H.B., Hansen, K.M., Kyng, M., Manikas, K.: Analysis and design of software ecosystem architectures—towards the 4S telemedicine ecosystem. *Inf. Softw. Technol.* **56** (11), 1476–1492 (2014)
14. Boucharas, V., Jansen, S., and Brinkkemper, S.: Formalizing software ecosystem modeling. In: *Proceedings of the 1st International Workshop on Open Component Ecosystems* (2009)
15. Cataldo, M., Herbsleb, J.D.: Architecting in software ecosystems: interface translucence as an enabler for scalable collaboration. In: *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pp. 65–72. ACM (2010)
16. Baars, A., Jansen, S.: A framework for software ecosystem governance. In: Cusumano, M. A., Iyer, B., Venkatraman, N. (eds.) *ICSOB 2012. LNBIP*, vol. 114, pp. 168–180. Springer, Heidelberg (2012)
17. Van Angeren, J., Kabbedijk, J., Jansen, S., Popp, K. M.: A Survey of associate models used within large software ecosystems. In: *IWSECO@ ICSOB*, pp. 27–39 (2011)
18. Pettersson, O., Svensson, M., Gil, D., Andersson, J., Milrad, M.: On the role of software process modeling in software ecosystem design. In: *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, pp. 103–110. ACM (2010)