Introduction

Slinger Jansen

The intricate networks of relationships between companies in the software industry have been the subject of much research and debate. The software ecosystems that can be identified within these networks are hubs of societal and technical innovation, strategic alliances leading to competitive edges, and powerhouses for co-creation of value. Software ecosystems touch everyone’s lives: from online social networks and their myriads of third-party plug-ins, to everyone’s pocket and the miniature network device in them, loaded with apps and nifty features. This book presents novel and innovative contributions from top scholars from around the world.

The most visible examples of software ecosystems are those that form around highly successful software platforms, such as Microsoft’s operating system, Apple’s and Google’s mobile phone operating systems, business platforms from Oracle and SAP, and the Facebook social network platform. Hordes of third-party developers have tried, with varying success, to use the tremendous customer bases and technical opportunities to create value for themselves and for the whole ecosystem. These third-party players have successfully created new innovations, tied different ecosystems together and created niche products that are extremely valuable to submarkets, that could never have been created by the platform leader.

The book is another step for the lively field of software ecosystems research. It is the outcome of an ambitious global collaboration, assembling the work of 39 authors, affiliated with 16 universities, in 15 countries, over five continents. The 15 chapters provide a nuanced overview of the research domain of software ecosystems and the implications they hold for the future of the software industry and the innovation thereof. The book offers (1) guidelines on how to analyze software ecosystems, (2) methods for managing and growing software ecosystems, (3) methods on transitioning from a closed software organization to an open one, and (4) instruments for dealing with open source, licensing issues, product management, and app stores in the context of software ecosystems. It is unique in bringing together industry experiences and academic views in this young field, tackling such challenges as the definition of fundamental concepts of software ecosystems, describing those forces that influence software ecosystems development and lifecycles, and the provision of methods for the governance of software ecosystems. The research highlights that made it into the book are chapters that have each proven to be at the forefront of ecosystems research, provide a novel contribution at the edge of what is known, have applied a solid research approach and collected data in a repeatable manner, and most of all, provide a state of the art overview for researchers to ground their future research.
1 WHY THIS BOOK?

While in the early days of software engineering a software product was the result of effort of an independent software vendor, modern software strongly relies on components and infrastructure of various third party vendors or open source consortia (Carmel, 1995; Sawyer, 2000; Cusumano, 2004). The relationships between software development firms and service companies shaped the product software landscape into software ecosystems where suppliers, resellers, implementers, and developers of software products, components and technologies collaboratively create competitive value. One could state that the success of a product software company therefore no longer depends on its own development quality but also on the way it manages its relationships with partnering organizations Hoch (2000).

In the cases described above there is one strong platform leader that, through opening up some aspects of its platform, maintains a strong hold on the development and growth of the ecosystems. It is not surprising that when looking at the other end of that scale, i.e., extremely open organizations, software ecosystems flourish as well. Open source consortia, such as Eclipse and Ubuntu, but also commercial organizations creating financial gains from these open platforms, are on the rise and have changed the markets for content management systems, integrated development environments, gaming and 3D engines, and many other domains. Both the commercial software producing organizations and the open source organizations are at the core of software ecosystems: both play central roles in large value networks and can influence the success of their ecosystems.

The first challenge that ecosystem researchers and software producing organizations run into is how to define their software ecosystem, or a software ecosystem in general. What constitutes a player in the ecosystem and what kind of powers do these players have? What do typical ecosystems look like? These questions are answered in the first part of this book. It can help researchers and managers to form concrete concepts of what ecosystems look like and what components make up an ecosystem.

The second question is how these software ecosystems can thrive. What powers does the platform leader have to grow the ecosystem or optimize the value that can be gained from the ecosystem? Why does a niche player choose a certain software ecosystem? What are barriers to entry and how do they influence the growth of an ecosystem? How is quality managed when several contributors, who are not under the control of the platform leader, can influence the total platform experience? And how do technical and functional characteristics of a software platform influence the shaping of a software ecosystem? These questions and fundamentally the question of how software ecosystems can thrive and what managers can do to govern and steer their ecosystem to success are answered in part two of this book.

Finally, the third question is how software ecosystems must be modeled and analyzed. By looking closely at the participants in an ecosystem and the gathering of such data, future software ecosystems researchers can create new theory from current data sets. Part 3 of this book for instance discusses how open source ecosystems can be studied by outsiders and what models can be used to visualize such data in a useful manner. Examples are also given of how particular databases, such as patent databases, can be used to visualize software ecosystems.

Readers could question whether the study of business ecosystems is sufficient. But
software is different, as many have confirmed. Software differs from physical goods in several ways. Software has no physical limitation, therefore the main limitations are conceptual, social and economical (Beizer, 2000; Messerschmitt and Szyperski, 2003). Much like other information products, software has a gross profit margin of 99 percent on sold products (Cusumano, 2004). In other words, reproduction costs for software products are next to zero (Postmus et al, 2009). Productivity gain resulting from the use of software is almost impossible to measure (Brynjofsson, 1993) and software quality assurance has more similarities with reviewing a large part of a law library or the construction of a skyscraper than it has with quality control of a TV set or mobile phone (Beizer, 2000).

Finally, no other industry can be compared to the software industry in terms of the influence exerted by the open source movement.

2 OBJECTIVES OF THIS BOOK

We had several reasons for compiling this book. Even though the editors are all involved in the organization and steering of the International Workshop on Software Ecosystems (2007–2012) and the International Conference on Software Business (2010–2012), it is hard to point to what precisely constitutes the software ecosystems research field, without having to point to all the proceedings from these events. Also, several books are available, such as “Platforms, Markets and Innovation” (Gawer, 2009), “Software Ecosystem” (Messerschmitt and Szyperski, 2003) (who have written the foreword to this book), and of course “The Keystone Advantage” (Iansiti and Levien, 2004), however, these bodes do not provide a comprehensive overview for use by software ecosystem researchers and educators. The logical next step was to bring together prime contributors from the research domain to create this book.

The second reason for the existence of this book is to inform business managers of the innovations that are taking place in the software ecosystems field, so that they may profit from the findings of the works that are compiled here. Many of the innovations actually come from the field and are still not applied broadly enough to move the industry forward. Therefore the editors, with some pride, wish to offer managers of software producing organizations this book as a source of inspiration for the growth of their innovativeness and success.

A third reason for creating this book lies in the fact that the software ecosystems research community is widely spread over several different communities, such as governance, software business, entrepreneurship, digital ecosystems, information systems, and many other related communities. The gathering of researchers with different backgrounds has been the best way to further build the growing software ecosystems research community. Finally and mainly, the aim of the book is to present a unified overview of the research from this interdisciplinary team of authors.

3 HOW THIS BOOK WAS COMPOSED

The aim of this book was to cover at least (1) defining software ecosystems and its research field, (2) the management of complex business networks that shape around
Software ecosystems

software and software platforms, and (3) the modeling and analysis of software ecosystems. Because this book is a compilation of chapters from the state of the art of software ecosystems, the topics that are discussed in each of the parts were inspired by a systematic study of the ecosystems domain, as presented in Chapter 4. The main research domains that were selected from the study are software evolution and co-innovation, business (network management), open source, and architecture. The three topics and five research domains created an overview for this book.

Based on the overview, authors were invited to write a topical chapter at the crossroads of their expertise, their earlier work, one of the main themes, and any of the research domains mentioned above. Furthermore, authors were asked to stress empirical results and present case studies and surveys to illustrate to readers what software ecosystems are. All invited authors were warned that chapters might be rejected if they were not innovative enough and after review one chapter was rejected due to extensive overlap with one of the other chapters.

Topics that were specifically excluded were agents and artificial intelligence in digital ecosystems, business ecosystems without a strong software component, and the history of large ecosystems. The book provides an overview of the state of the art of software ecosystems research but in no way claims to be complete for two reasons. First, the current state of the art is developing rapidly and new works may have come out during the course of publishing of this book. Second, we have selected the contributions with the vision of ecosystem analysis, modeling, management, and governance in mind, which may exclude other areas. I also firmly believe that several books can be written about the role of software repository mining in software ecosystems, but in this book we have only addressed this topic in some of the chapters.

4 CONTENT OF THIS VOLUME

In each of the following chapters a relevant business, technology, or mixed issue is treated using a software ecosystem view of the software industry and the surrounding business networks. The book is divided into three parts. Part 1 deals with the complex issue of unifying different conceptual fundamentals of software ecosystems into one and illustrates this with literature, case studies, theory, and examples. Part 2 is concerned with the management and governance of software ecosystems. The main topics handled in this part are openness of software producing organizations, partnership models, and quality management. Finally, in Part 3 the topics of ecosystem modeling, visualization, data bases and data gathering, and analysis of ecosystems are discussed.

In the first chapter entitled “Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance”, Michael Cusumano and I explore several definitions of software ecosystems and try to establish a consensus of the different definitions that exist which is so far lacking. Closer exploration shows that the definition provided in an earlier work by Jansen, Finkelstein, and Brinkkemper (Jansen et al, 2009) encompasses several other definitions. The operationalization of the definition is also further explored with examples, to enable readers to correctly identify boundaries of ecosystems. The chapter further discusses governance modes for both commercial and open source organizations that have (some) control over the underpinning technology for
Introduction

Chapter 1, entitled “A Software Ecosystem and its Components: Understanding the Ecosystems’ Association” presents the features of ecosystems and their components, i.e., the software supply network level of individual organizations, the software ecosystem level, and the software ecosystems level that observes relationships between ecosystems. Furthermore, two case studies are presented to illustrate the concerns of coordinators of software ecosystems. Although the work is not as fundamental as Chapter 1, it goes deeper into software ecosystems and provides researchers with typical exploration paths for future work. It also enables researchers to position their work on the different scope levels.

Chapter 2, entitled “Business Network Management as a Survival Strategy” presents the three scope levels of software ecosystems, i.e., the software supply network level of individual organizations, the software ecosystem level, and the software ecosystems level that observes relationships between ecosystems. Furthermore, two case studies are presented to illustrate the concerns of coordinators of software ecosystems. Although the work is not as fundamental as Chapter 1, it goes deeper into software ecosystems and provides researchers with typical exploration paths for future work. It also enables researchers to position their work on the different scope levels.

Chapter 3, entitled “Guiding Principles of Natural Ecosystems and Their Applicability to Software Ecosystems” settles the point of whether it is useful to compare software ecosystems to their natural counterparts. The operationalization of the comparison is a relevant starting point for future research and identifies issues and potential research directions for future work. After the operationalization, the chapter presents a framework for a sustainable software ecosystem and points specifically at (1) the identification and enhancement of knowledge cycles, (2) fostering of diversity of contributors and users, (3) the definition and monitoring of health indicators, (4) decision making across organizational boundaries, (5) creating an infrastructure for fostering social interaction within the ecosystem, and (6) creating adaptable artifacts, processes, and stakeholders. The authors further strengthen their points using highly illustrative case studies of Eclipse, Apple’s iOS, and Google’s Android operating system. The theory presented in this chapter effectively complements the theory presented in the two preceding chapters.

In Chapter 4, entitled “A Systematic Mapping Study on Software Ecosystems through a Three-dimensional Perspective” we make a first exhaustive exploration of the literature in software ecosystems from three perspectives (business, technical, and social) and this work has been fundamental to the creation of this book. The application of a systematic mapping study to such a young field has proven successful for two reasons. First, the study shows that ecosystem literature focuses on eight topics: (1) operating systems, (2) software architecture, (3) modeling of software ecosystems, (4) open source, (5) software product lines, (6) business issues, (7) software evolution, and (8) software co-innovation. Second, the chapter identifies relevant research directions for software ecosystems in the fields of open source ecosystems, governance, ecosystem analysis, platform openness, software architecture, and software quality. In compiling this book we have attempted to cover each of these areas, although there appears to be a lack of work in the areas of software modeling, co-innovation, and software evolution in the perspective of software ecosystems. Chapter 4 further strengthens the point that more research is required in the field of software ecosystems (Jansen et al, 2009).

Chapter 5, entitled “Managing Software Ecosystems through Partnering” describes one of those issues that is discussed extensively in the industry literature but still lacked a formal view: partnering models for software platforms. These partnering models have been established by many of the platform leaders (Microsoft, SAP, Apple, Google, etc.) but had still to be generalized in a formal model. The authors present a conceptual overview that attempts to capture all aspects addressed by such a partnering model and can be
used as a foundation for the creation of new partnering models. The chapter emphasizes the power organizations can exert with these partnership models and illustrates how business models are influenced by them. Furthermore, entry barriers that can withhold niche players from joining an ecosystem are defined, to show owners of partnership models that they may be inadvertently rejecting niche players that could strengthen the ecosystem. This chapter is a prime example of ecosystems research currently taking place and we encourage others to similarly uncover essential business issues and demonstrate their relationships to the formation and growth of software ecosystems.

Chapter 6, entitled “The Challenge of Heterogeneously-licensed Systems in Open Architecture Software Ecosystems”, addresses the issue of heterogeneous licensing over different products and components and the license conflicts that may arise from this. Software ecosystems companies that reuse other components will frequently run into this underestimated problem. The chapter provides an approach for analyzing, establishing, and resolving such conflicts. It provides advice to software producing companies on how to handle such conflicts and is perhaps one of the more practical chapters in the book for managers that work within software ecosystems. It is therefore recommended reading for researchers with an interest in heterogeneous licensing and business managers in software producing organizations.

In Chapter 7, entitled “Framing Management Practices for Keystones in Platform Ecosystems”, a structured overview is presented of four management practices that support platform integrators operating in software ecosystems. The four practices are technology scouting, orchestration, software supply network management, and technology asset management. These management practices are linked to the technology and innovation management processes, which may help to systematically respond to challenges in software ecosystems. With the use of an illustrative case the authors show how a software ecosystem view can be used by a platform integrator to further their innovation goals. This chapter is highly relevant for those new to software ecosystems, but are sure they are in one and wish to make better informed management decisions.

Chapter 8, entitled “Architectural Openness: Comparing Five Mobile Platform Architectures” is the result of some of the most interesting research into software architecture. By comparing the software architecture of five mobile operating systems and surveying a number of developers that develop these ecosystems, the authors show that openness is only one of the many reasons why software developers choose to join a specific ecosystem. Another finding is that the openness of certain architectures is directly related to the strategy that is followed by the platform leader. The mobile ecosystems that are discussed are RIM’s BlackBerry, Apple’s iOS, Google’s Android, Nokia’s Symbian, and Microsoft’s Windows Mobile. Finally, an openness model is provided for mobile operating systems that could potentially steer developers of such operating systems to either close or open certain parts of their architecture in compliance with their strategy.

In Chapter 9, “The Open Software Enterprise Model: How Open is my Software Business?” a fine-grained model is presented that enables a software producing organization to establish the openness of their organization and their product. The “openness score” that follows from the model can be used to objectively compare organizations. The model has the specific aim of reducing the subjectivity with which software producing organizations are judged to be open or closed, without any objective way of measuring this. With the use of four case studies, contrasts are shown between the openness of
organizations previously considered open or closed. In the future, we hope this model will be adopted by others in making software product procurement decisions, or in antitrust cases.

Chapter 10, entitled “Quality Review and Approval Methods for Extensions in Software Ecosystems” performs a survey of the quality review and approval methods that are applied within software ecosystems, with the aim of increasing quality when the platform leader does not have full control over the contributions of its niche players. The chapter provides some prime examples of software producing organizations that have successfully managed software ecosystems in extremely open environments. Furthermore, it is shown that software producing organizations are not using many of the freely available methods for the quality improvement of contributions. This chapter is especially suitable for those who can exert control over an ecosystem and wish to increase the quality of contributions of third parties.

Chapter 11, entitled “Measuring the health of a business ecosystem” continues with a discussion on the health of ecosystems. Using the foundations provided by Iansiti and Levien (Iansiti and Levien, 2004) of robustness, niche creation, and productivity, the authors create an operationalization of health measures. The operationalization can be used to establish what participants in the ecosystem are actively contributing to the health of the ecosystem and which ones need to be encouraged or excluded from the ecosystem. The operationalization is evaluated using a quantitative analysis of the largest IT ecosystems in The Netherlands. This chapter is relevant for ecosystem coordinators and platform leaders who want to further strengthen the relationships with their most powerful partners and for those who generally wish to increase the health of their ecosystem.

Chapter 12, entitled “Analysing Ecosystems for Open Source Software Developer Communities”, presents an extensive framework enabling researchers to study open source developer communities and the ecosystems in which they thrive. Insights gained by such empirical studies will help tool developers to provide better automated support, and project managers to use guidelines and best practices that improve the work practices of such communities. The Gnome ecosystem is taken as a validating case study and the authors show that the Gnome community is actually divided over a number of sub-communities. Also, the authors show that the activity in the Gnome ecosystem is decreasing and also specializing in subdomains of the Gnome features. The most important contribution is perhaps the elegance with which the authors retrieve data from open data sets and then thoroughly analyze these data.

Chapter 13, “Open Source Ecosystems: a Tale of Two Cases”, describes the challenge of creating open source communities for industrial software that was originally developed as closed source but later made its source code open source and available to a large community of developers. Supporting processes, guidelines and best practices are discussed and illustrated both from the software and the business perspective through industrial cases taken from a Finnish context. The popularity of moving towards more open ecosystems by commercial software producing organizations makes this chapter a timely one. The chapter sheds light on a relevant new phenomenon that specifically targets the growth of a new ecosystem that may or may not develop, depending on the process that is followed in opening the ecosystem.

Chapter 14, entitled “The Technological Roadmap of Cisco’s Business Ecosystem”, takes the company Cisco as a starting point. By studying patent data and co-authors of
patents with Cisco, the author has developed an interesting perspective on the growth of Cisco and its ecosystem strategy. By relating the patent co-author data to developments in mergers and acquisitions of Cisco, a vivid picture is painted of the way in which Cisco tends to its ecosystem. Implications of symbiosis, platform, and co-evolution are provided for managers to challenge the contemporary business environment. The chapter is also relevant to researchers, as it provides an interesting research method to study company case studies.

Finally, in Chapter 15, entitled “Unraveling Ruby Ecosystem Dynamics: A Quantitative Network Analysis”, a case study is presented of the open source ecosystem called Ruby. This ecosystem, which has very low (if any) coordination, has grown explosively over the last couple of years. The main conclusions from the case study are (1) the Ruby software ecosystem consists of developers, gems and relationships and (2) developers within the software ecosystem fulfil several distinctive roles, each of different value to the ecosystem and also (3) within the software ecosystem most activity is caused by only a small part of the ecosystem. The top 90 percent of the open source components used in Ruby development has been developed by only 10 percent of the total number of open source contributors. The value of this knowledge lies in deciding how to better manage or steer software ecosystem governance. Trying to lure additional developers to your ecosystem in order to expand your ecosystem may not be the best way of managing a software ecosystem; motivating existing developers to work together more on existing gems is a better way to get a solid and healthy ecosystem. This chapter together with Chapter 13 provide a relevant starting point for researchers wishing to research open source ecosystems and creating visualizations of large ecosystems.

5 CONCLUDING REMARKS

Although this book provides the first systematic overview of this research field, the topic still consists mainly of unchartered territory. In that sense, the book truly is a snapshot of what we know now. Fortunately, each of the chapters contains one or more research challenges that may direct future researchers to new research questions and unexplored data sets. The editors therefore hope that researchers find inspiration in the book for their own future endeavors. Enjoy the book!

BIBLIOGRAPHY


