Theoretical foundations of software ecosystems

Geir K. Hanssen[1], Tore Dybå[1]

[1] SINTEF, NO-7465 Trondheim, Norway
{Geir.K.Hanssen,Tore.Dyba}@sintef.no

Abstract. Software ecosystems have become a defined and active field of research based on the recent emergence of new open business models leading to new roles and patterns for collaboration, innovation, and value proposition. In this paper we look into the theoretical foundations of this new research field. We review recent publications to describe the present theorization; we discuss the relevance of theory and, in particular, we argue for the relevance of socio-technical theory and the related theory of organizational ecology in the continuing research on software ecosystems. To summarize we define a preliminary theoretical framework to guide and support future research.

Keywords: Software ecosystems, theory

1 Introduction

'Software ecosystems' is becoming a new and important field of research fueled by new business models in the software engineering domain, representing a redefinition of traditional roles and patterns for collaboration and innovation. This creates complex networked communities of organizations or actors. Such communities are often based on a common interest in a central software technology, like a software product line or platform that offers opportunities of value increasing technologies and services by a variety of actors [22]. We see cases where a keystone organization dominates the development [24]; in other cases, control is diffuse, like in open source communities [35]. Such cross-technology and cross-organizational networks, which we define as software ecosystems, represent a radical shift in how software engineering is being done, influencing fundamental aspects such as control, collaboration, business models, and innovation [39, 54]. In short, software engineering is becoming an open process in a complex distributed environment.

Following this trend in industry, we also see the emergence of a research community sharing an interest in software ecosystems and how they affect the software engineering discipline. A dedicated international workshop series is established as well as various publications at international conferences and in scientific journals, including a special issue on software ecosystems [6]. So far, the terminology and definitions within this research community vary greatly – the concept of software ecosystems is rather vague and diverse and it can be hard to see how results and contributions from empirical studies relate to each other beyond the research of single or close groups of authors. This is natural as a novel field of

research develops, but this diversity also imposes a challenge for the joint development of this research field and the knowledge on software ecosystems. In order to address this challenge we believe that it is important to develop a common theoretical foundation to advance and direct the development of the research field and the empirically grounded knowledge it produces.

A theory can be seen as generalized knowledge about a discrete topic or phenomenon. According to Gregor [21], a theory may serve to i) analyse, ii) explain, iii) predict, iv) explain *and* predict, or v) prescribe. There is no unified definition of how theories are constituted but often they are made up of *constructs* and the *relationships* between these [47]. Such theories may be formed in at least three ways [ibid.]: i) theories from other disciplines may be used as they are, ii) theories from other disciplines may be adapted to software engineering before use, and, iii) theories may be generated from scratch.

Based on recent studies and our understanding of the fundamental features and challenges related to software ecosystems we find socio-technical theory [49] and the related theory of organizational ecology [18, 48] particularly relevant as they address fundamentally important concepts such as a control, self-regulation, network organization, the role of technology, and the sharing of values [22]. We also believe that there exist other theories that can be related to this in order to develop a common theoretical basis for the software ecosystem research field.

The aim of our paper is threefold. First, we explain the role of theory building in shaping a novel research field and show how this may enable the development of a common knowledge base and how it can guide and coordinate further research. Second, we investigate the state of research on software ecosystems in order to build an overview of theorizing in this emerging research field. Finally, we point to existing theories, both from the software ecosystems literature as well as other relevant domains, which we find particularly relevant to the specific challenges imposed by the emergence of software ecosystems.

2 Research approach

In order to build an overview of the state of the research on software ecosystems and in particular the present focus on theorizing, we applied the following seven-step approach to identify relevant contributions:

1. We used the Web of Science index to identify potentially relevant studies by searching for ‘software ecosystems’ in the Topic-field. The search was restricted by setting the ‘subject area’-category to ‘computer science’. The resulting 468 titles were evaluated and 27 were found to be relevant. We selected titles that were considered to address software ecosystems. After a review of the abstracts we kept 16 titles.
2. We browsed the proceedings of the International Workshop on Software Ecosystems (IWSECO) for the whole lifetime of the workshop (2009-2011). We selected only full papers and found 3 from 2009, 11 from 2010 and 7 from 2011.

3. We included the five publications from a recent JSS special issue on software ecosystems [6].
4. Books on software ecosystems were identified using Amazons search facilities. Two titles explicitly addressing software ecosystems were identified – one by Messerschmitt and Szyperski (2003) [39], and one by Popp and Meyer (2010) [46].
5. To complement the collection, we consulted a recent systematic mapping study by Barbosa and Alves [3] that identified 44 relevant publications (whereof one thesis and two books). Out of these we had found 16 from steps 1-4, the 28 missing studies were downloaded.

In total, we identified and collected 72 publications, which at the time of writing is the most comprehensive collection of publications directly relating to software ecosystems.

6. All 72 publications were reviewed to identify cases of theorizing. We found 40 of these to show signs of theorizing, some with an explicit focus – others more peripheral.
7. From the review we built a structured overview of which theories that are addressed in this literature (see section 4).

From this overview and our own recent theory development [22] we propose a theoretical framework for this new research field (see section 5).

3 Software ecosystems as a trend and defined research area

Software ecosystems is a recent term, referring to a networked community of organizations or actors, which base their relations to each other on a common interest in the development and use of a central software technology. Some other definitions of this emerging concept have been proposed, for example by Jansen et al. [33]: "*a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them*" (p. 2). Another definition by Bosch [5], focusing more on the common interest in the software and its use, is: *"the set of software solutions that enable, support and automate the activities and transactions by the actors in the associated social or business ecosystem and the organizations that provide these solutions"* (p. 2).

Well known examples of communities that may be seen as software ecosystems are Apples iPhone/Appstore platform and the open-source development environment Eclipse. The first is an example of a partially closed and controlled ecosystem, and the latter is an example of an open ecosystem allowing more flexibility in use and development. This simply illustrates that the ecosystem concept may refer to a wide range of configurations. Yet, they all involve two fundamental aspects; a network of organizations or actors, and a common interest in the development and use of a central software technology. These organizations may have different relations to the central software technology, and for this reason, different roles in the ecosystem. In

our definition of the concept, there are at least three key roles. First, one organization (or a small group) acts as the *keystone organization,* and is in some way leading the development of the central software technology. The second key role is the *end-users* of the central technology, who need it as a tool to carry out their business, whatever that might be. The third key role is *third party organizations* that use the central technology as a platform for developing related solutions or services. In addition to these key roles, various other related roles might be part of the ecosystem, for example standardization organizations, resellers, operators, and others.

The emergence of software ecosystems relates to the inherent potential for open innovation [19], increased involvement of users [40], wider and faster market impact [39], and opportunities of increased profit [46]. Interaction in open and cross-organizational networks may create increased value in the intersection between roles. For example, being a third-party provider of solutions or services based on a central platform means efficient access to both an enabling technology and an already established, known, and accessible user group. For a keystone organization, this in turn may lead to increased market shares and respective sales, increased feedback, and a more dedicated focus on the core technology as external actors focus on value adding services.

These examples motivate increased research effort to develop a better understanding of how software ecosystems work and how various actors can reap benefits. We argue that theorizing is an important vehicle in this development.

4 Signs of theorizing

We browsed the 72 identified publications looking for theorizing, for example, references to established theories in order to motivate a study or to develop the background for a study. We also looked for uses of theory in order to analyze or explain findings as well as publications that seek to develop theory. By *theory* we refer to both explicit theories such as for example *socio-technical theory*, which is well established, but also to wider concepts such as *organizational ecology* and *open innovation*. Following this we present a review of studies and their theoretical orientation. To form an overview, we developed higher-order themes for the identified theories or concepts [11].

Table 1. Theorizing in the software ecosystems (secos) literature

Theories and concepts	*Used in*	*Theoretical focus*
Functions and features of ecosystems		
Openness and transparency	[1, 9, 29, 32, 35, 41]	Understanding Secos as open systems
Innovation	[2, 4, 28, 52]	Explaining the innovation in Secos
Control and the keystone role	[2, 22, 24, 29, 32, 46, 51, 52, 53]	Explains how Secos are controlled – in some cases by a central organization
Performance and economics	[10, 39, 44, 45, 46]	Explains performance of Secos, e.g. revenue mechanisms
Activity theory*	[50]	Explains the interaction of human activity and consciousness within its relevant environmental context
Transaction cost theory*	[23, 38]	Explains coordination of distributed value creation activities
Structuring and shaping of ecosystems		
Roles and relationships	[27, 30]	Explains the roles in software ecosystems and their relationships
Theory of design rules (systems theory) *	[37]	Understands a system through the interdependence between its components
Socio-technical theory*	[14, 22]	Understanding Secos as the interplay between the social system and the technical system
Intermediary theory*	[23]	Explains the need and function of intermediaries to provide services in the Secos
Evolution of Secos	[22, 25]	Describes how Secos evolves
Describing and modeling ecosystems		
Networking and network visualization	[4, 15, 20, 25, 26, 29, 31, 34, 37, 39, 55]	Describing and visualizing Secos as a network
Ecology inspired	[8, 13, 24, 46, 51]	Understanding Secos as a variant of a natural ecosystem
Modeling and taxonomies	[5, 7, 22, 28, 43, 46, 51, 55]	Describe Secos through models and components

From this initial overview we see that most focus (simply measured in number of publications) is on various functions and features of software ecosystems. We find all of these highly relevant and would like to point out that all of these concepts and

theories are associated with large knowledge bases and vital research traditions. It is beyond the scope of this workshop paper to provide detailed insights; we leave this for future work. We have also found quite many contributions describing software ecosystems as models and visual representations. Such descriptions do not necessarily reflect key features and mechanisms of ecosystems but are important in order to achieve a more thorough understanding of ecosystems. Further on, we see that there are relatively few publications that explicitly refer to established theories (marked with an asterix). These also does not relate to each other. We see this as a sign of infancy and expect future concurrent studies to relate to each other and a common theoretical focus and that this will mature the theoretical basis over time [16].

In addition to contributions found by explicitly looking for the 'software ecosystem'-keyword we are also aware of other streams of research that we consider to be highly relevant. Cusumano and Gawer[12] develops the concept of *platform leadership* – which is directly relevant to understand the role of keystone organizations and similar. Eisenmann et al. [17] and others discusses *two-sided markets* where two (or more) groups of end-users take complementary roles and gain network benefits through their relationship with a central platform. A simple example of a two-sided market is credit card platforms, joining byers and sellers. This stream of research is relevant to develop the understanding of the central software technology.

5 Towards a theoretical foundation

We seek to combine the theorizing in the present literature and connect it with our recently proposed theoretical framework [22], to present a nascent but cohesive overview. Our initial suggestion is based on the concept of organizational ecology, originally defined by Trist [48]. We have derived and adapted five propositions from Trists work, addressing what we believe are fundamental aspects of software ecosystems:

i) Member organizations in a software ecosystem are linked to a key organization among them, which acts as **a central referent organization**, doing so even though **members** are only partially under its control or linked to it only through **interface relations**. We see that control moves from the supplier towards other roles.
ii) Software ecosystems promote **self-regulation** through frequent interaction and feedback between the actors.
iii) Software ecosystems have a **networked character** representing radically different structures compared with traditional business models.
iv) Software ecosystems exist through the use of **information and communication technology (ICT)**. Such ICT use is fundamental to achieve development, marketing, communication, coordination, deployment of solutions, and economic transactions.

v) Software ecosystems exhibit **shared values,** which constitute the motivation of the various actors. The shared values consist of the central software product or platform as well as the business domain itself.

We find that some of the contributions we have identified in section 4 directly relate to and potentially extends this theoretical understanding of software ecosystems: Descriptions of roles and relationships and the focus on the particular role of a central organization relate to proposition i). Studies focusing on theory of design and open systems theory may develop proposition ii) and explain *how* self-regulation works. Studies reflecting socio-technical theory are potentially valid for all propositions, but in particular iv) in order to better understand the role of the technical components in a socio-technical system. Finally, we also believe that studies that interpret software ecosystems as networked organizations will develop proposition iii), potentially improving the understanding of this new community level, spanning multiple (traditional) organizations. We also find contributions on (open) innovation and transparency to be relevant to proposition iii). Finally, we see that proposition v) that explains the function of a shared value among the actors in a software ecosystem is *not* present in any of the theoretical contributions we have identified so far (with an exception of our own study [22]).

Further on, we have identified several concepts and theories in the present literature that is *not* covered by, or directly related to our set of propositions: Theorizing on performance, transactions, and revenue mechanisms is highly interesting and could potentially explain some of the motivation and reasons behind the emergence of software ecosystems and also be relevant in order to understand software ecosystems as self-regulated systems. Also, we find the reference to activity theory relevant as it potentially may explain how work and business in software ecosystems affect human activity. We have also found several studies aiming to develop a better understanding of roles and relationships, which is highly important to better understand how software ecosystems function. The same goes with attempts to model and visually represent ecosystems. Finally, we have also found studies that theorize on the emergence, development, and evolution of software ecosystems.

6 Concluding remarks and future work

The overview of theorizing in the software ecosystems research literature presented in this paper constitutes the most complete summary, so far. We have identified quite many relevant concepts and theoretical traditions from a variety of research fields beyond software engineering. Building a joint theoretical basis for this research field is an extensive task and will have to span multiple and various studies over time. We hope that our contribution can be seen as a starting point and that it will serve as an invitation to the software ecosystems research community to continue theorizing. We strongly believe that this will help the community to better plan future studies as well as to better relate our common interest and knowledge on software ecosystems. After all, *"Nothing is as practical as a good theory" (K. Lewin, p. 129) [36].*

We advice future work to further and deeper investigate, develop and relate the *content* of the theories and fundamental concepts that we have identified so far, both those that align with our proposed framework but definitively also those that are not covered. Furthermore, we also believe there may be valuable knowledge and inspiration to find in related and more established research fields such as business ecosystems and even natural ecosystems [42]. It is however important to develop this knowledge based on validated empirical studies of the *software* business as it displays radically different characteristics than other businesses. Examples are the short distance from design to use, the intangibility of software, the high innovation velocity, among others. .

References

1. Anvaari, M. and Jansen, S. Evaluating Architectural Openness in Mobile Software Platforms. In proceedings of 4th European Conference on Software Architecture (ECSA'10). 2010. Copenhagen: Springer: p.85-92.
2. Arndt, J. and Dibbern, J. Co-Innovation in a Service Oriented Strategic Network. In proceedings of International Conference on Services Computing (SCC'06). 2006. Chicago: IEEE: p.285 - 288
3. Barbosa, O. and Alves, C. A Systematic Mapping Study on Software Ecosystems. In proceedings of Third International Workshop on Software Ecosystems. 2011. Brussels, Belgium: CEUR: p.12.
4. Basole, R.C. and Karla, J., *On the Evolution of Mobile Platform Ecosystem Structure and Strategy*. Business & Information Systems Engineering, 2011. **3**(5): p. 313-322.
5. Bosch, J. From Software Product Lines to Software Ecosystems. In proceedings of 13th International Software Product Line Conference (SPLC'09). 2009. San Fransisco, USA: IEEE Computer Society: p.111-119.
6. Bosch, J., *Software Ecosystems Taking Software Development Beyond the Boundaries of the Organization*. Journal on Systems and Software, 2012. **85**(7): p. 1453–1454.
7. Boucharas, V., Jansen, S., and Brinkkemper, S. Formalizing Software Ecosystem Modeling. In proceedings of First International Conference on Open Component Ecosystems (IWOCE'09). 2009. Amsterdam: ACM: p.41-50.
8. Briscoe, G. and de Wilde, P. Computing of Applied Digital Ecosystems. In proceedings of International Conference on Management of Emergent Digital EcoSystems (MEDES'09). 2009. Lyon: ACM: p.28-35.
9. Cataldo, M. and Herbsleb, J.D. Architecting in Software Ecosystems: Interface Translucence as an Enabler for Scalable Collaboration. In proceedings of Second International Conference on Software Ecosystems in cooperation with the European Conference on Software Architecture. 2010. Copenhagen, Denmark: ACM.

10. Ceccagnoli, M., Forman, C., Huang, P., and Wu, D.J., *Cocreation of value in a platform ecosystem: the case of enterprise software*. MIS Quarterly, 2012. **36**(1): p. 263-290.
11. Cruzes, D.S. and Dybå, T., *Research synthesis in software engineering: A tertiary study*. Information and Software Technology, 2011. **53**(5): p. 440-455.
12. Cusumano, M.A. and Gawer, A., *The Elements of Platform Leadership*. MIT Sloan Management Review, 2002. **43**(3): p. 51-58.
13. Dhungana, D., Groher, I., Schludermann, E., and Biffl, S. Software Ecosystems vs. Natural Ecosystems: Learning from the Ingenious Mind of Nature. In proceedings of Second workshop on Software Ecosystems in conjunction with 4th European Conference on Software Architecture. 2010. Copenhagen, Denmark: ACM.
14. dos Santos, R.P. and Wernes, C.M.L. Revisiting the Concept of Components in Software Engineering from a Software Ecosystem Perspective. In proceedings of 4th European Conference on Software Architecture (ECSA'10). 2010. Copenhagen: ACM: p.135-142.
15. Dreyfus, D., Iyer, B., Venkatraman, N., and Lee, C., *Dual Networks of Knowledge Flows: An Empirical Test of Complementarity in Software Ecosystems*, in *(Whitepaper)*. 2005. p. 51.
16. Edmondson, A.C. and McManus, S.E., *Methodological fit in Management Field Research*. The Academy of Management Review, 2007. **32**(4): p. 1155-1179.
17. Eisenmann, T., Parker, G., and Van Alstyne, M.W., *Strategies for Two-Sided Markets*. Harvard Business Review, 2006. **84**(10): p. 2-11.
18. Emery, F.E. and Trist, E.L., *The Causal Texture of Organizational Environments*. Human Relations, 1965. **18**: p. 21-32.
19. Enkel, E., Gassman, O., and Chesbrough, H., *Open R&D and open innovation: exploring the phenomenon*. R&D Management, 2009. **39**(4): p. 311-316.
20. Fricker, S. Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems. In proceedings of First International Workshop on Software Ecosystems. 2009. Milan, Italy: CEUR-WS: p.19-33.
21. Gregor, S., *The Nature of Theory in Information Systems*. MIS Quarterly, 2006. **30**(3): p. 611-642.
22. Hanssen, G.K., *A Longitudinal Case Study of an Emerging Software Ecosystem: Implications for Practice and Theory*. Journal on Systems and Software, 2012. **85**(7): p. 12.
23. Hilkert, D., Wolf, C.M., Benlian, A., and Hess, T. The "As-a-Service"-Paradigm and Its Implications for the Software Industry – Insights from a Comparative Case Study in CRM Software Ecosystems. In proceedings of FIrst International Conference on Software Business. 2010. Jyväskylä, Finland: Springer: p.125-137.
24. Iansiti, M. and Levien, R., *Strategy as Ecology*. Harward Business Review, 2004(March): p. 1-12.

25. Iyer, B., Dreyfus, D., and Gyllstrom, P., *A Network-based View of Enterprise Architecture*, in *Handbook of Enterprise Systems Architecture in Practice*, P. Saha, Editor. 2007, IGI Global: Singapore. p. 500.
26. Iyer, B., Lee, C., and Venkatraman, N., *Managing in a "Small World Ecosystem": Some Lessons from the Software Sector*. California Management Review, 2006. **48**(3): p. 28-47.
27. Iyer, B. and Venkatraman, N. The Changing Architecture Of Global Work: Opportunities And Challenges. In proceedings of the Keane Workshop on Global Work. 2006. the Keane Workshop on Global Work: (Whitepaper): p.20.
28. Janner, T., Schroth, C., and Schmid, B. Modelling Service Systems for Collaborative Innovation in the Enterprise Software Industry - The St. Gallen Media Reference Model Applied. In proceedings of 2008 IEEE International Conference on Services Computing (SCC'08). 2008. Hawaii: IEEE: p.145-152.
29. Jansen, S., Brinkkemper, S., and Finkelstein, A., *Providing Transparency In The Business Of Software: A Modeling Technique For Software Supply Networks*, in *Establishing the Foundation of Collaborative Networks*, L. Camarinha-Matos, et al., Editors. 2007, Springer: Boston. p. 677–686.
30. Jansen, S., Brinkkemper, S., and Finkelstein, A. Component assembly mechanisms and relationship intimacy in a software supply network. In proceedings of 15th International Annual EurOMA Conference. 2008. Groningen: Own: p.10.
31. Jansen, S., Brinkkemper, S., and Finkelstein, A. Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems. In proceedings of First International Workshop on Software Ecosystems (IWSECO). 2009. Falls Church, USA: CEUR-WS: p.34-48.
32. Jansen, S., Brinkkemper, S., Souer, J., and Luinenburg, L., *Shades of gray: Opening up a software producing organization with the open software enterprise model*. Journal on Systems and Software, 2012. **85**(7): p. 1495-1510.
33. Jansen, S., Finkelstein, A., and Brinkkemper, S. A sense of community: A research agenda for software ecosystems. In proceedings of 31st International Conference on Software Engineering (ICSE'09). 2009. Vancouver, Canada: IEEE Computer Society: p.187-190.
34. Kabbedijk, J. and Jansen, S. Steering Insight: An Exploration of the Ruby Software Ecosystem. In proceedings of 2nd International Conference on Software Business (ICSOB'11). 2011. Brussels: Springer: p.44-55.
35. Kilamo, T., Hammouda, I., Mikkonen,, T., Aaltonen, T., *From proprietary to open source—Growing an open source ecosystem*. Journal on Systems and Software, 2012. **85**(7): p. 1467–1478.
36. Lewin, K., *The research centre for group dynamics at Massachusetts Institute of Technology*. Sociometry, 1945. **8**(126-135).
37. Liu, X., Lee, C., and Iyer, B. The Impact of Design Moves on Platform Adoption: The Case of Microsoft Windows OS. In proceedings of 39th Hawaii International Conference on System Sciences (HICSS'06). 2006. Hawaii: IEEE: p.1-10.

38. McGregor, J.D. A Method for Analyzing Software Product Line Ecosystems. In proceedings of 4th European Conference on Software Architecture. 2010. Copenhagen: ACM: p.73-80.
39. Messerschmitt, D.G. and Szyperski, C., *Software Ecosystems, Understanding an Indespensable Technology and Industry*. 2003, Cambridge: The MIT Press. 424.
40. Messerschmitt, D.G. and Szyperski, C., *Marketplace Issues in Software Planning and Design*. IEEE Software, 2004. **21**(3): p. 8.
41. Molder, t. Clopenness of Systems: The Interwoven Nature of Ecosystems. In proceedings of Third International Workshop on Software Ecosystems (IWSECO'11). 2011. Brussels: Springer: p.52-64.
42. Moore, J.F., *Predators and prey: A new ecology of competition*. Harvard Business Review, 1993. **71**(3): p. 75-86.
43. Petterson, O., Svensson, M., Gil, D., Andersson, J., and Milrad, M. On the Role of Software Process Modeling in Software Ecosystem Design. In proceedings of 4th European Conference on Software Architecture (ECSA'10). 2010. Copenhagen: ACM: p.103-110.
44. Popp, K.M. Goals of Software Vendors for Partner Ecosystems – A Practitioner´s View. In proceedings of First International Conference on Software Business (ICSOB'10). 2010. Jyväskylä: Springer: p.181-186.
45. Popp, K.M. Hybrid revenue models of software companies and their relationship to hybrid business models. In proceedings of Third International Workshop on Software Ecosystems (IWSECO'11). 2011. Brussels: ACM: p.77-88.
46. Popp, K.M. and Meyer, R., *Profit from Software Ecosystems*. 2010, Norderstedt: Herstellung und Verlag: Books on Demand. 238.
47. Sjøberg, D.I.K., Dybå, T., Anda, B.C.D., and Hannay, J.E., *Building Theories in Software Engineering*, in *Advanced Topics in Empirical Software Engineering*, F. Shull, J. Singer, and D.I.K. Sjøberg, Editors. 2007, Springer Verlag: Heidelberg.
48. Trist, E.L., *A Concept of Organizational Ecology*. Australian Journal of Management, 1977. **2**(2): p. 161-175.
49. Trist, E.L., *The evolution of socio-technical systems*, in *Occasional Paper No. 2*. 1981, Quality of Working Life Centre: Toronto. p. 67.
50. Uden, L. and Damiani, E. ActivityTheoryforOSS Ecosystems. In proceedings of 2007 Inaugural IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2007). 2007. Cairns, AUstralia: IEEE: p.223-228.
51. van den Berk, I., Jansen, S., and Luinenburg, L. Software Ecosystems: A Software Ecosystem Strategy Assessment Model. In proceedings of 4th European Conference on Software Architecture (ECSA'10). 2010. Copenhagen: ACM: p.127-134.
52. Ven, K. and Mannaert, H., *Challenges and strategies in the use of Open Source Software by Independent Software Vendors*. Information and Software Technology, 2008. **50**(9-10): p. 991-1002.
53. Viljainen, M. and Kauppinen, M. Software Ecosystems: A Set of Management Practices for Platform Integrators in the Telecom Industry. In

proceedings of 2nd International Conference on Software Business. 2011. Brussels: Springer: p.32-43.
54. von Hippel, E., *Democratizing innovation*. 2005, Cambrige: MIT Press.
55. Yu, E. and Deng, S. Understanding Software Ecosystems: A Strategic Modeling Approach. In proceedings of Third International Workshop on Software Ecosystems (IWSECO'11). 2011. Brussels: CEUR: p.65-76.