

# Scientists' Needs in Modelling Software Ecosystems

Slinger Jansen  
Utrecht University, the  
Netherlands  
slinger.jansen@uu.nl

Eko Handoyo  
Utrecht University, the  
Netherlands  
eko.handoyo@gmail.com

Carina Alves  
Federal University of  
Pernambuco  
cfa@cin.ufpe.br

## ABSTRACT

Currently the landscape of software ecosystem modelling methods and languages is like Babel after the fall of the tower: there are many methods and languages available and interchanging data between researchers and organizations that actively govern their ecosystem, is practically impossible. The lack of a universally accepted set of modelling methods is hampering the advancement of software ecosystems research. Using a literature study and a set of interviews amongst peers, we aim to establish a set of understandings and requirements for a universally accepted set of software ecosystem modelling methods. The work is an initial push in a larger research initiative that has the goal of advancing the maturity of (software) ecosystems modelling. The success of such an initiative will be found in the availability of common databases, better interchange formats between researchers, and more capable software ecosystem modelling tools.

## 1. INTRODUCTION

The software ecosystems research appears to be at a high point. Since the literature study of Manikas and Hanssen [1] the field seems to have exploded and it is almost impossible to keep up with all the new research that is coming out in the domain. It is time for the domain to mature. This is achieved by advancing a number of issues that have not been taken care of yet in the research agenda of Jansen et al. [2]: software ecosystem management practices, architecting in software ecosystems, the effects on software development of software ecosystems, and finally software ecosystem modelling, which is the topic of this paper.

The need for advancement in the domain of software ecosystem modelling is observed by many. Barbosa and Alves identify eight major fields within the software ecosystems domain, and one of them is software ecosystem modelling [3]. The aforementioned work of Manikas and Hanssen also identifies software ecosystems modelling as a strategic theme. We identify at least three important uses of software ecosystems modelling. First, it is the most important way to gain understanding [4] about software ecosystems: whether these are open source, commercial, hybrid, or simulated ecosystems. Secondly, we believe that analysis is best done

through modelling [5, 2, 6]. Thirdly, we believe prediction of how the ecosystem develops pending certain decisions is best done by formal modelling of software ecosystems [7, 8].

We use the previously established definition of software ecosystems: a set of organizations functioning as a unit and interacting with a shared market for software and services, together with the relationships amongst them [2]. When we address software ecosystem modelling, we mean to address the (formal) methods used to model, analyse, visualise, and simulate software ecosystems. The visualizations created are typically used for providing insight and for visual analysis. These techniques are, however, insufficient when a more in-depth analysis is necessary. As such, we also take into account further analysis methods, such as statistical analysis methods that typically borrow heavily from (social) network analysis. Figure 1 displays some examples of successful ecosystem visualizations. Obviously, there are many modelling and visualization methods available. Despite these advancements, none of them have tried to make the requirements of software ecosystem modelers explicit. This is surprising, as the community would benefit from a standardized way of exchanging data, creating models, and modelling ecosystems.

Currently the landscape of software ecosystem modelling languages is like Babel after the fall of the tower: there are many languages available and interchanging data between researchers and organizations governing their ecosystems is practically impossible. The main research question investigated in this paper is: "What are the needs of researchers for software ecosystems modelling"? In Section 2, we describe the research method: a literature study and in-depth interviews with selected software ecosystem researchers about the current state of ecosystem modelling and the limitations that current modelling languages have. Using the data from these interviews, we present that the main reasons for modelling ecosystems are insight, analysis, and comparison in Subsection 3.1. In Subsection 3.2 we elaborate what is typically modeled: organizations (actors), relationships, and flows. Thirdly, in Subsection 3.3 is illustrated how researchers typically model software ecosystems, which is mainly through social network models, goal models, and supply chain models. In Section 4 we outline some of the largest challenges in ecosystems modelling, such as large data sets, lacking features in modelling tools, and the lack of research in software ecosystem evolution. We conclude with a set of plans for the future in Section 5: the creation of a research agenda on ecosystem modelling, the extension of existing tools, and perhaps even a data sharing infrastructure for software ecosystems data.

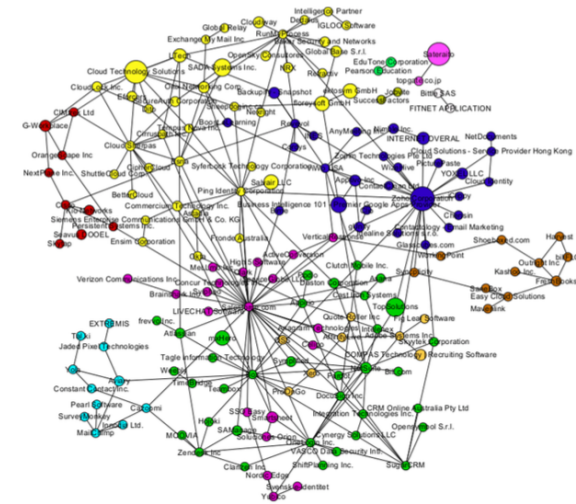
## 2. RESEARCH METHOD

The research goal has been to explore the domain of software ecosystem modelling. As this is an exploratory task, two ap-

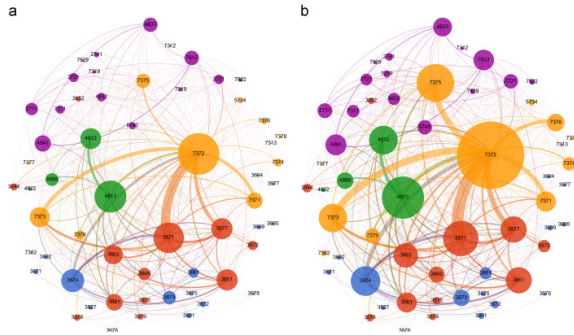
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IWSECO/WEA 2015

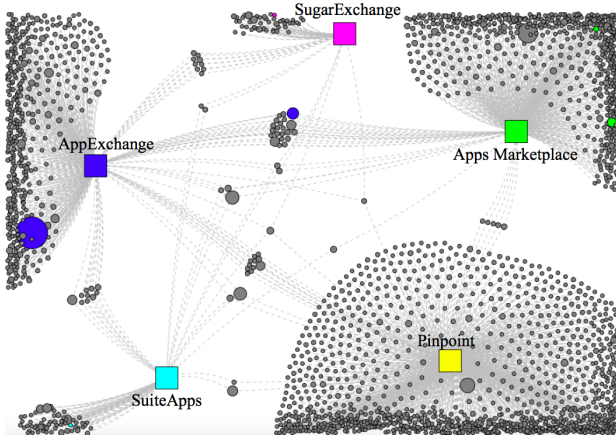
Copyright 15 ACM X-XXXXX-XX-X/XX/XX ...\$15.00.



(a) van Angeren [9]: Cluster visualization of the Google Apps ecosystem. The node sizes are representative for the number of apps included in the app store. Nodes with similar color belong to the same cluster as assigned by the modularity algorithm.



(b) Basole [10]: Illustrating convergence within the ICT industry through clustering.



(c) Burkard [11]: The squares are business apps market places. The circles are business apps, which in many cases are multi-homed, i.e., available in multiple app markets.

Figure 1: Some examples of software ecosystem model visualisations.

proaches have been followed: that of (a) literature study and (b) interviews with experts in the domain of ecosystem modelling have been conducted.

The study was carried out using a qualitative research approach. Qualitative research aims to investigate and understand social and cultural phenomena in the context where they exist. Furthermore, qualitative research aims to place the studied phenomena in a holistic framework and suggests that social and cultural phenomena are best investigated by studying people's actions in and verbalized thoughts about the social context in which they act [12]. Qualitative research methods are useful when the purpose is to explore an area of interest, to obtain an overview of a complex area, and to discover diversities and variety rather than similarities [13]. It is also preferable to use a qualitative approach when the aim is to improve the understanding of a phenomenon where little is known.

## 2.1 Step I: Literature study

Literature studies use a more rigorous and well-defined approach to reviewing the literature in a specific subject area. Literature studies are used to answer well-focused questions about scientific practice. For the research in this paper we explicitly looked for papers from the study of Manikas and Hanssen that included software ecosystem visualizations. As the interviews progressed we also were confronted with the opportunity to include more works, mostly cited by the interviewees, that were inspirational to the interviewees' publications. We chose to include those publications as well in our research.

## 2.2 Step II: Interview study

*Planning* - The first phase of the research elaborated a brainstorming and planning meeting to identify different fields of interest and to plan the study. We aimed at a variety of modelling frameworks, with respect to languages, layout algorithms, visualization tools, mathematical methods.

*Data Collection* The study uses a semi-structured interview strategy [13]. The experts were recruited by inviting authors from the publications found in the literature study. The six (6) interviews were attended by one interviewee and one interviewer. All interviews varying in length from 30 to 65 minutes, were recorded. Transcripts of all interviews were made in order to facilitate and improve the analysis process. The transcripts varied between 3 and 8 pages in size. A trial interview was organized between the authors to finetune the questions. The six interviewees were found pragmatically and were originally asked because of their contributions to the workshop on software ecosystems. However, as we received overwhelming positive replies (very few declined) to our invitations, we stopped after six interviews and did not continue to approach the list of potential interviewees.

*Analysis* - The content analysis involved marking and discussing interesting sections in the transcripts. A topic map has been created of the content and is shown in figure 2. Please note not all relationships have been mapped out yet, due to time constraints.

*Final analysis of all data* - The process throughout the three steps of the study reached what can be described as theoretical saturation. Apart from one more category, the third step mainly provided more detail and confirmed previous findings. The artefacts used for the grounded theory analysis can be requested from the authors. Please note that the tools were specifically not addressed in this study. At the time of writing the authors are also involved in exhaustively studying the tools used for creation of software ecosystem modelling and visualization.

## 3. RESULTS

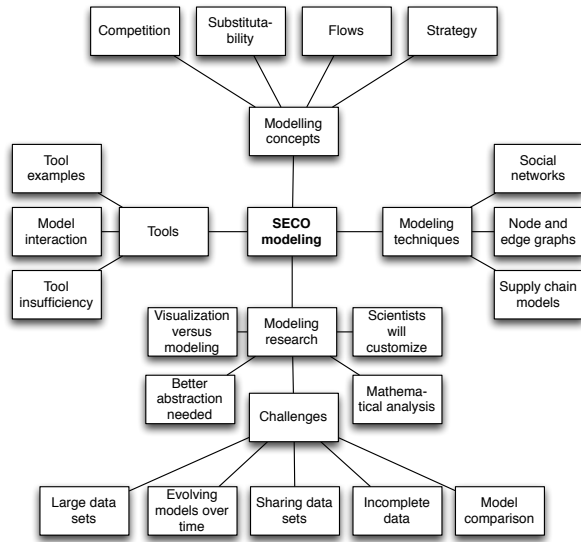


Figure 2: The topic map for the study. Comments were marked with these terms, as to further enable analysis of groups of comments surrounding the same topics.

Both the literature study and the interviews have been conducted with the aim of discovering why visual models of software ecosystems are made, what their objectives are, and how they can potentially be supported with new approaches. In the following paragraphs we discuss the findings of the literature study and the interviews, according to several subquestions. Please find an overview of the results of the literature study in Table 1.

The researchers agree that there is no good standard language available yet for software ecosystem researchers. They frequently compare the models to supply chain modelling and social network analysis. The consensus is that, although these tools are powerful and supportive of software ecosystem analysis, there still exist opportunities for improvement.

### 3.1 Why model ecosystems?

The modelling objectives have been extracted from the publications, but also from the interviews. We find that the following modelling objectives are most common: providing insight, analysis of static ecosystems, and ecosystem comparison. We address each of them here.

1. **Providing insight** - Most researchers cite that a picture in a densely written paper on ecosystems can effectively provide an overview of what the ecosystem looks like. The motivations that are most commonly given are those of face validity, easy identification of key players in ecosystems, such as keystone players, and providing insight into density of relationships.
2. **Analysis of static ecosystems** - Another use of visualizations is doing actual analysis of the ecosystem. It becomes possible to identify key relationships (bridgers within the ecosystem, for instance), relationship density and connectedness within the ecosystem, and player size differences. Typically, the reader is taken through the same steps as the researcher, to provide some kind of analytic insight into the ecosystem. A major part of such research is the identification

of key elements in the ecosystem, ranging from identifying CEOs [16] to architectural control points [19].

3. **ecosystem comparison** - One of the more exotic research objectives is the comparison of ecosystems [22] through visualization. It is beneficial for the analysis of ecosystem maturity, development, and dynamics, to look at how shapes, connectivity, and objects in the ecosystem models differ.

*II:* “you want to work on more than one ecosystem visualization at a time, there is limited mean for comparison, rather than visual inspection or just putting this matrix to each other.”

### 3.2 What to model?

The models and visualizations of the researchers under study are varied. We observe simple arrow and box models, supply-chain like supply models, and most frequently so-called dot-clouds: models of nodes and edges that typically show actors and relationships between them. The concepts underlying these models can be very different: for some, the nodes are software companies whereas with others these are components, open source developers, or even patents. We observe the following elements in the ecosystem models:

1. **Organizations and their types** - In most of the papers the entity around which the model revolves is the organization, whether it is banks in the mobile payments ecosystem [17], phone operators in the mobile ecosystem [14], or niche players in any other. An important aspect is that many modellers take the organization type into account, by varying in node shape, size, and color.
2. **Relationships** - An essential entity to even make an ecosystem is the relationship. In most of the encountered works relationships are a fundamental part of the model. *II:* “I was looking at business ties between companies.” Relationships can be component dependencies, trade relationships, and collaborations. Relationships can vary in thickness, to indicate tightness, or in shape, to indicate relationship type. One of the most essential points made in the interviews about relationships is that they are rich and typically concerned with the technical and business perspective. *I2:* “I would like to capture the technical relationships like API dependencies, but also non-technical interactions, like cash flows.”
3. **Flows** - An entity that does not receive enough attention are flows across relationships. Questions about f.i. the amounts of code that flow from one open source project to another, the amounts of money flowing from a software reseller to its platform provider, and the coupling tightness between two software libraries are insufficiently studied. *I2:* “We build on top of social networks because they are inherently good at flows, like knowledge flows and dependencies.”

The models and visualizations of those models under study are not good at providing insight into other factors. The interviewees mention explicitly software architecture, substitutability, and the multiplicity of relationships. *II:* “if your scope pertains the more platform architecture related stuff, ... it is very difficult to do with social network analysis.” Secondly, the models are also criticized for not dealing well with dynamic and evolutionary aspects of ecosystems. *I3:* “It’s about scale, about distributed control. If you want the concept of evolution in it, you need an analogy to species, whether that is software, relationships.”

### 3.3 How to model?

Authors state in the interviews that the modelling objective is perhaps the most important in determining the modelling method, the elements that will be modeled, and the data that will be used for modelling. **I4**: “*It all depends on what you need.*”. Furthermore, **I1** states that “*most of the time people simply pick a definition of software ecosystems that suits their needs, and work from there.*” The following methods are proposed most often and also observed in the literature:

- 1. Social networking models** - Currently, social networks are seen as a fitting overlay for software ecosystems. These people networks closely resemble the relationships, flows, and organisational entities of software ecosystems. However, these models are considered inadequate for modelling complex flows, extremely large networks, and software architecture. **I2**: identified that “*Social networking tools, though highly capable, are insufficiently capable at handling substitutability.*” SECO researchers will want to know, for instance, whether a certain software component is irreplaceable in a specific ecosystem, as that gives the provider of that component an essential role in the ecosystem and a competitive advantage, because it enables the provider to extract value from the ecosystem. Another problem with social network graphs is that they insufficiently capture value flows and relationship multiplicity between participants. **I4**: “*It is also really hard to model both competition and collaboration in a software ecosystem using social network analysis.*”
- 2. Goal modelling languages like i\*** - Yu’s i\* has been applied in software ecosystems [24] by some, although there are few works illustrating this well. Goal modelling is mentioned explicitly because it is capable of expressing the goals of organizations within an ecosystem, effectively combining competition and collaboration, for instance. The interviewees recommend that i\* is somehow simplified to accommodate both the scale and complexity of software ecosystems.
- 3. Supply chain networks** - With the techniques of Jansen et al. [25] surrounding software supply chains, researchers have been successful at conceptual modelling of ecosystems. The technique, however, has been insufficiently adopted for the modelling of larger complex ecosystems, based on quantitative work. This could be caused by a lack in tooling or formalisation. The interviewees are generally positive about such models. **I5**: “*I like supply chain model not only from code level, but also from an organizational perspective because it gives a really nice structure for [looking at] up-stream versus down-stream.*” **I6**: “*If these were organizations in a supply chain there might be another graph that represents the trust between organizations, so there are technical relationships and business types relationships, such as trust and reliability.*”

The techniques presented here provide an interesting dilemma for any software ecosystem researcher. It is obvious that most researchers choose to implement a method that best fits their requirements and almost always need to customize the method at hand. An extra note that must be made is the role of formalisation. When a method is used for automated reasoning or data analysis, authors generally choose for some kind of formalisation. **I4**: “*I am currently dealing with a complex problem that simply requires formalization.*”

As mentioned by Eriksson et al. [26], combining language and mathematics would be more efficient and meaningful. One interviewee mentioned “I suspect the mathematical methods or interpretations are very abstract.” Another said “*I’m not aware of the mathematical methods ... it depends on the purpose, what kind of information you want to express.*” Some express that more examples and guides are required in this domain. **I6**: “*The application of networking metrics [to our node and edge graphs] is one of the hardest things to do ... the tools don’t support you in deciding what metrics to use [for analysis]. We had to educate ourselves. It would be nice if the tool made more suggestions about that.*”

## 4. MODELLING CHALLENGES

Several modelling challenges are addressed by the interviewees. First, large data sets are identified as a challenge. **I1**: “*Yes, I think there is limitation in the modelling large networks. [Our tools] after certain capacity started slowing down’ [and performing worse]*”.

Secondly, interviewees indicate that the visualizations of models typically scale poorly. **I4**: “*These pictures are really messy, it’s better to group into clusters.*” The complaints are in both directions: zooming out is complicated because clustering hides too much information. Zooming is also hard, because the tools are typically not designed to provide deeper insight into the artefacts and data that underly the models. **I6**: “*The tools are bad at this too, I’d like to bore down and zoom in and out. I want to go specifically deeper into files, I don’t want the boxes bigger, I want to see what’s inside the boxes. The data that we get is very rich. There’s code inside the files we’re studying.*” **I6**: “[in regards to] visualization we typically looked at it at a smaller scale for reasons of practicality.”

Thirdly, the interviewees see large data sets as a curse and a blessing. On the one hand tools deal poorly with them, information is lost when clustering large data sets, and researchers are forced to scope their research. **I5**: “*I think we could probably be constraining the ecosystem, even that meaning you only get a partial view of the ecosystem.*” Large data sets are typical for the domain. Whether it is the thousands of Ruby developers collectively creating software libraries or the more than a million apps in the mobile app stores: data is relatively freely available. This is useful, because it enables us to “*triangulate missing data.*” (**I1**). It must be noted that although large data sets are available, data about commercial companies is generally seen as hard to obtain. **I1** again: “*we invested significant efforts into gathering our data.*”

Finally, one particular aspect of modelling that each interviewee expresses interest in but has little experience in, is the study of ecosystem evolution and dynamics. Researching ecosystems’ development over time is found extremely challenging, especially when the static views already have plenty to offer. It is suggested that the tools should be more dynamic: **I6**: “*If the diagrams were more dynamic, they could be more useful.*” Furthermore, it is suggested that ecosystem evolution should be studied over time: **I3**: “*If you want the concept of evolution in it, you need an analogy to species, whether that is software or relationships.*” **I6** agrees with the use of this: “*It would be great if we could show general visualizations over time.*”

## 5. CONCLUSIONS

This paper tries to answer basic questions of modelling ecosystems: why model ecosystems, what should be modeled, and how should it be modeled. The questions are answered in part through literature study and interviews with publishing authors in the domain. Secondly, a number of challenges of modelling are identi-

fied.

In regards to future work, we plan to develop this preliminary paper into a research agenda for software ecosystem modelling, through a further analysis of the results. We aim to do a more extensive analysis of the data, using the topic map as a guideline. Along the themes found in these topic maps, we intend to create specific research themes and perform more detailed literature studies in these domains. A final outcome of such a project could be the development of a shared database for ecosystem data. Furthermore, we are aiming to study (and perhaps) extend ecosystem modelling tools, to assess for which purposes they are most appropriate. Also, we aim to develop methods and tools for the study of evolving ecosystems over time. Finally, there are opportunities to study ecosystem boundaries and their effect on study results.

## Acknowledgments

This research project has been supported and financed by the Indonesian Ministry of Education and Culture, Directorate General of Higher Education PhD scholarship grant number: 3374.1/E4.4/2011. Furthermore, we'd like to thank the interviewees, for their time and their infinite wisdom. Finally, we thank the Great Chinese Firewall. If it hadn't been for continuously hampered internet traffic and a prolonged stay in China of one of the authors, this paper may never have come to fruition.

## 6. REFERENCES

- [1] K. Manikas, K. M. Hansen, Software ecosystems-a systematic literature review, *Journal of Systems and Software* vol. 86 (2012) pp. 1294–1306.
- [2] S. Jansen, A. Finkelstein, S. Brinkkemper, A sense of community: A research agenda for software ecosystems, in: *Software Engineering-Companion Volume*, 2009. ICSE-Companion 2009. 31st International Conference on, IEEE, 2009, pp. 187–190.
- [3] O. Barbosa, C. Alves, A systematic mapping study on software ecosystems, in: *Proceedings of the workshop on software ecosystems*, 2011.
- [4] R. C. Basole, T. Clear, M. Hu, H. Mehrotra, J. Stasko, Understanding interfirm relationships in business ecosystems with interactive visualization, *Visualization and Computer Graphics*, IEEE Transactions on 19 (12) (2013) 2526–2535.
- [5] I. Hunink, R. van Erk, S. Jansen, S. Brinkkemper, Industry taxonomy engineering: the case of the european software ecosystem, in: *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*, ACM, 2010, pp. 111–118.
- [6] T. Janner, C. Schroth, B. Schmid, Modelling service systems for collaborative innovation in the enterprise software industry-the st. gallen media reference model applied, in: *Services Computing*, 2008. SCC'08. IEEE International Conference on, Vol. 2, IEEE, 2008, pp. 145–152.
- [7] K. Karhu, T. Tang, M. Hämäläinen, Analyzing competitive and collaborative differences among mobile ecosystems using abstracted strategy networks, *Telematics and Informatics* 31 (2) (2014) 319–333.
- [8] K. Karhu, A. Botero, S. Vihavainen, T. Tang, M. Hämäläinen, A digital ecosystem for co-creating business with people, *Journal of Emerging Technologies in Web Intelligence* 3 (3) (2011) 197–205.
- [9] J. van Angeren, V. Blijleven, S. Jansen, S. Brinkkemper, Complementor embeddedness in platform ecosystems: The case of google apps, in: *Digital Ecosystems and Technologies (DEST)*, 2013 7th IEEE International Conference on, IEEE, 2013, pp. 37–42.
- [10] R. C. Basole, H. Park, B. C. Barnett, Coopetition and convergence in the ict ecosystem, *Telecommunications Policy*.
- [11] C. Burkard, T. Draisbach, T. Widjaja, P. Buxmann, Software ecosystems: Vendor-sided characteristics of online marketplaces, *Informatik* 2011.
- [12] M. D. Myers, D. Avison, Qualitative research in information systems, *Management Information Systems Quarterly* 21 (1997) 241–242.
- [13] C. Robson, *Real word research*, Oxford: Blackwell, 2002.
- [14] R. Basole, J. Karla, On the evolution of mobile platform ecosystem structure and strategy, *Business & Information Systems Engineering* 3 (5) (2011) pp. 313–322.
- [15] R. C. Basole, M. A. Bellamy, Visual analysis of supply network risks: Insights from the electronics industry, *Decision Support Systems* 67 (2014) 109–120.
- [16] R. C. Basole, J. Putrevu, On leadership, alliance formation, and enterprise transformation, *Journal of Enterprise Transformation* 4 (1) (2014) 28–50.
- [17] B. Iyer, Ecosystems: A method for visualizing software ecosystems, *Proceedings of 4th International Workshop on Software Ecosystems* (2012) pp. 1.
- [18] S. Yu, C. J. Woodard, Innovation in the programmable web: Characterizing the mashup ecosystem, in: *Service-Oriented Computing—ICSOC 2008 Workshops*, Springer, 2009, pp. 136–147.
- [19] D. Dreyfus, B. Iyer, Managing architectural emergence: A conceptual model and simulation, *Decision Support Systems* 46 (1) (2008) 115–127.
- [20] F. Mödritscher, W. Taferner, A. Soylu, P. De Causmaecker, Visualization of networked collaboration in digital ecosystems through two-mode network patterns, in: *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, ACM, 2011, pp. 158–162.
- [21] J. van Angeren, S. Jansen, S. Brinkkemper, Exploring the relationship between partnership model participation and interfirm network structure: An analysis of the office365 ecosystem, in: *Software Business. Towards Continuous Value Delivery*, Springer, 2014, pp. 1–15.
- [22] J. van Angeren, C. Alves, S. Jansen, Analyzing complementor interactions in commercial platform ecosystems: The role of governance mechanisms, *Journal of Systems and Software*, submitted.
- [23] J. Kabbedijk, S. Jansen, Steering insight: An exploration of the ruby software ecosystem, in: *Software Business*, Springer, 2011, pp. 44–55.
- [24] E. Yu, S. Deng, Understanding software ecosystems: A strategic modeling approach, *proc of 3rd IWSECO* (2011) 65–76.
- [25] V. Boucharas, S. Jansen, S. Brinkkemper, Formalizing software ecosystem modeling, in: *Proceedings of the 1st International workshop on Open component ecosystems*, ACM, 2009, pp. 41–50.
- [26] O. Eriksson, B. Henderson-Sellers, P. J. Ågerfalk, Ontological and linguistic metamodeling revisited: A language use approach, *Information and Software Technology* 55 (12) (2013) 2099–2124.

Table 1: Software Ecosystems Modeling Methods

Pub	Research objective	Ecosystem	Research approach	Data source	Network metric	Unit of analysis	Tool	Network topology	Layout algorithm	Final data set
[14]	To identify, analyze, visualize the mobile platform ecosystem	iOS, Android, LiMo, webOS, Blackberry	Data are integrated, seeded into three segments & visualized	Thomson Reuters SDC Platinum, Connexiti database	Network centralization, Network density, Average degree of nodes	Mobile device manufacturer, Mobile network operator, Mobile platform provider	Pajek	Scale-free	KamadaaÅKawaFruchtermanÅReingold	20 companies, 200 relationships
[10]	To identify, assess, visualize the ICT ecosystem	Hardware component, Hardware equipment, Software, Telecommunication, Media	Data are collected and extracted into five segments and visualized	Thomson Reuters SDC Platinum	Coopetition, Complexity, Velocity	Equity& others, Licensing, Manufacturing-OEM-supply chain, Technology transfer, R&D-marketing, Joint ventures	Gephi	Random	Noverlap	20,232 firms, 20,870 alliances
[15]	To identify, evaluate and mitigate risk of electronics industry	Electronics industry	Data are compared, validated, constructed into three tiers & visualized	Thomson Reuters Financial SDC Platinum Alliance & Joint-Venture, Connexiti database	Betweenness centrality, Financial risk, Inventory variability	Firm, Network	Gephi	Scale-free	Concentric	911 firms, 7,311 relationships
[16]	To determine the top-performing CEOs	Automobile, Consumer goods, Energy, Financial services, Healthcare, Industrials, IT & telecom, Materials, Retail, Utilities	Data are collected, extracted into five segments & visualized	100 best-performing CEOs, Thomson Reuters SDC Platinum	Average degree, Average weighted degree, Density, Average clustering coefficient, Average path length	Strategic, R & D, Supply & Manufacturing, Marketing, Licensing, Technology Transfer, Cross Border, Joint Venture	Gephi	Small-World	OpenOrd	-
[4]	To design,develop an interactive visualization tool	Mobile	Data are integrated, classified into fourteen segments, normalized & visualized	Thomson Reuters SDC Platinum, Capital IQ Compustat	Compositional, Segmentation, Connectivity (Path, Segment, ScatterNet, Geography)	Mobile network operator, Internet service provider, Gaming provider, Cable provider, Photography & digital imaging, Media & entertainment provider, Content provider, Silicon vendor & other component provider, Device manufacturer, Network infrastructure provider, Platform provider, Application software provider, Service & billing provider, System integrator	dotlink360	Circle-relative	-	2,809 companies, 1,7025 agreements
[17]	To draw, analyze ecosystem	Mobile payment	Data are determined, identified the properties, generated input file, visualized & interpreted	deLicio.us	-	Technical, Marketing, Licensing	Pajek	Random	-	-
[11]	To analyze characteristics of five online marketplaces	Online marketplaces	Data are collected,extracted into five marketplaces, visualized	Salesforce.com, Google Apps, Pinpoint, SugarCRM, NetSuite	-	AppExchange, Apps Marketplace, Pinpoint, SugarExchange, SuiteApp.com	-	Cellular	-	-
[18]	To investigate, visualize the web 2.0	Software	Data are collected, extracted into two tiers, visualized	ProgrammableWeb	Mean degree, Normalized degree, Network density, Characteristic path length, Clustering coefficient	Mashup, API	NetDraw	Core-periphery	NetDraw's node repulsion	2664 mashups, 590 APIs
[19]	To define, identify, show of the architectural control points	Financial services	Data are collected, extracted into two tiers, visualized	FinServ	-	Software components, Data dependencies	-	Small-World, Random	-	158 nodes
[20]	To examine, analyze, visualize patterns in networked collaboration	Wiki platforms	Data are collected, extracted into three collaborations, visualized	SONIVIS, Wikiversity, Wikiquote	-	Contributors, Articles	-	Small-World	PALADIN	27,625 contributors, 19,213 articles, 158,271 relations
[21]	To investigate the extent to which participation in the partnership models on ecosystems structure	Office365	Data are collected from ecosystem hubs and by visiting the web sites of partners within the ecosystem	Microsoft Pinpoint Marketplace, CrunchBase	Size, Network density, Centralization, Modularity, Clustering coefficient	Complementors, Applications	Gephi	Small-World	Modularity	550 complementors, 1,204 applications, 787 relationships
[22]	To compare four ecosystems in terms of descriptive statistics, and unearth how governance mechanisms have helped shape the ecosystem	Google & Microsoft	Data is manually collected from ecosystem hubs & by visiting the web sites of partners within the ecosystem	Google Apps, Google Chrome, Microsoft Office365, Internet explorer	Size, Network density, Degree centrality, Centralization, Modularity, Clustering	Complementors, Applications	Gephi	Small-World	Modularity	3,602 complementors, 5,468 applications, 2,683 relationships
[23]	To analyze an ecosystem, provide descriptive data on the ecosystem, & provide advice to increase the health of an existing ecosystem	Ruby	Data are extracted from an ecosystem hub,analyzed	Ruby Community's official gem	Downloads, Yahoo hits, Size, Lines of code	Developers, Gems, Relationships	Gephi	Random	-	4,784 Developers, 10,046 Gems, 13,103 Relationships
[9]	To analyze, visualize of the Google Apps platform ecosystem	Google Apps	Data are obtained,extracted into two segments, visualized	Google Apps Marketplace	Size, Density, Centralization, Embeddedness, Eigenvector centrality, Clustering coefficient	Complementors, Applications	Gephi	Small-World	Modularity	992 complementors, 1341 applications