The Network Characteristics of Open Source Software Business – a Multi-disciplinary Case Study

Järvensivu Juha[1]; Helander Nina[2]
1 Researcher, Tampere University of Technology, juha.jarvensivu@tut.fi
2 Senior Researcher, Tampere University of Technology, nina.helander@tut.fi

Abstract—**Open Source has become an important phenomenon in the field of software business, as Open Source can be seen as a viable alternative for traditional, proprietary way to develop competitive software solutions. Open Source bases on a networked way to develop software – networks are formed within one specific Open Source project (i.e. internal network) and between several Open Source projects (i.e. external network). This paper presents a case study of an Open Source project called Laika and its external network formed with other related Open Source projects. The network analysis of Laika is carried out both from technological and business-oriented viewpoints and by following qualitative research methods.**

***Keywords*—Open Source software, business networks, communities, dependencies, case study.**

I. OBJECTIVES OF THE PAPER

This paper presents a multi-disciplinary approach to the phenomenon of Open Source software development, covering the issue of Open Source network characteristics both from technological and more business-oriented industrial network point of view. This paper has a strong empirical focus, as the aim of the paper is to analyze Open Source software network characteristics through an in-depth, qualitative case study of a one specific Open Source community, called Laika.

We have gathered data related to Laika both by discussing with the people involved in Laika project and by participative observation in Laika project. The network analysis carried out and presented in this paper is, however, only a snapshot of the Laika project and its external network structure.

Although this paper presents only a snapshot of Laika's external network, we believe that Laika offers an interesting field to address the issue of Open Source networks. In this paper, we will introduce and analyze the Laika network, through the elements of mutuality, interdependence, distance, priorities, different power relations, and investments made in the relationships. These elements are based on a literature analysis of both technology and industrial network approach fields.

We will also present in the paper the concept of "super-communities", which are collections of Open Source communities working for a common good.

Towards the end of the paper, we will discuss what kind of theoretical contribution the Open Source phenomenon can bring in our opinion and what kind of future research we will carry out in this research area.

II. TECHNOLOGICAL VIEWPOINTS TO OSS NETWORKS

Collaboration between Open Source communities is commonplace. A main idea of the Open Source is to utilize existing third party Open Source components and projects instead of doing all by oneself. This easily forms complex networks where actors are dependent on each others. Dependency can be one way or two way dependency depending on how communities see partner's achievements.

The basic principle for the cooperation is voluntary participation (Frees 2002). In contrast to the traditional industrial projects, communities dependent on each other have proceeded without any formal agreements. This type of collaboration is suitable only if actors see that the partner's action yield some benefit for them too. If partner's achievements are deemed useless, it is not worth participating in the partnership.

A close cooperation may also cause changes in priorities. As a result of collaboration, projects are utilizing more and more other's features, and connections between actors are becoming more and more complex. The traditional approach is that one's own project has always the highest priority. However, in the Open Source project it is sometimes more important to give support to another related project than to continue develop one's own project.

Another phenomenon typical for the Open Source project is that the communication between other Open Source communities is mostly handled by a public way. For the most part the communication is handled via mailing lists, irc-channels and other public manners such as using a bugzilla to report errors of the applications. The benefit of this kind of communication is that anyone who is interested of the project can join to the mailing list to get the latest news, and share own ideas and insights with other developers. As a result, new patches and releases for the open source projects have usually been published more often than in the case of proprietary software solutions (Porter 2001). This may cause some unexpected problems if an Open Source project is dependent on several other Open Source projects. If the other projects that the specific project is dependent on will be updated frequently, a lot of extra work has to be done to ensure the compatibility to new versions of the projects.

On the other hand the stability of Open Source projects is better than in the case of proprietary software. In the other words the project cannot be interrupted suddenly through a single actor because of large amount of independent developers.

III. INDUSTRIAL NETWORK APPROACH TO OSS NETWORKS

According to Easton (1992), the industrial network approach aims at achieving understanding of industrial markets as complex networks that are formed from a bunch of inter-organisational relationships. Möller & Wilson (1995) summarize that network theory aims at providing conceptual tools for analysing both structural and process characteristics of industries. The goal is to understand complex systems of relationships by studying an industry from a holistic perspective. They also point out that both the structural and process characteristics can be viewed at different levels, which are the industry level, the level of firm in industry, the level of the firm as a nexus of business exchange relationships, and the relationship level.

Håkansson & Snehota (1989) point out that the network approach takes into consideration the relations between different actors. All the actors, their activities, and resources are bonded, linked, and tied up together, and in this way they build up a wide network. Easton (1992) illustrates the basic elements of the network approach from four different viewpoints, or metaphors: networks as relationships, positions, structures, and processes.

A basic assumption with the network approach involves the essential unit of **relationships**, from which proceeds understanding of the network as a sort of cluster of relationships. Furthermore, relationships are characterised by four basic elements: mutuality, interdependence, different power relations, and investments made in the relationship. It is also important to keep in mind that the effects of the relationship can be both positive and negative, and both primary and secondary functions in the relationships can be found. Primary functions refer to the relationship's effects on the parties involved in the dyad, whereas secondary functions refer to the effects that the relationship has on the other actors in the network (Anderson, Håkansson, & Johanson 1994).

Analysis of networks as **positions** mainly involves examination of the network from the viewpoint of a single actor, which can be either an individual or an organization (Easton 1992). However, micro and macro network positions can be differentiated. Micro positions are characterised by the role of the actor in relation to another actor, the actor's significance to another actor, and the nature (strength) of the relationship between two actors. Thus, micro positions focus on dyadic relationships. A broader perspective, on the other hand, is characteristic of macro positions – e.g., also the nature of so-called indirect relationships and the company's own role in the overall network.

Networks as **structures** are concretised through the interdependencies between the actors. If there are no interdependencies between the actors, neither will there be any network structure. The greater the interdependence of the actors, the clearer the structure of the network. Thus, there can be so-called 'tight' and 'loose' networks. Tight networks are characterised by a great number of bonds between the actors and well-defined roles and functions for actors. Loose networks, on the other hand, are illustrated by the opposite characteristics. The question of the boundaries of the network is also related to the 'networks as structures' perspective. Although in principle the whole world economy could be seen as one huge network, it is essential for the purposes and implementation of research to divide networks into smaller pieces and examine these smaller parts of networks.

The nature of networks as **processes** mirrors the nature of the networks themselves: networks are stable but not static. Due to the interrelationships among actors in the network, evolutionary changes are more characteristic of networks than radical changes are (Easton 1992).

In next, we will apply these network analysis tools coupled with a more technical analysis in the case of Laika.

IV. THE CASE OF LAIKA IN TERMS OF NETWORK CHARACTERISTICS

Laika is an Open Source development project aimed at the creation of an integrated development environment for developing applications for embedded Linux devices that run on the Maemo platform (Laika 2006). The main idea of the project is to integrate the work of several Open Source projects in a single software tool (Järvensivu et al. 2006).

Although Laika itself forms an interesting network to study, in this paper we will address the external network of Laika, i.e. the network that Laika forms together with other Open Source projects. In fact, Laika is dependent on many other Open Source projects such as Maemo, Scratchbox, Eclipse and CDT, which form a network of dependencies outside the project.

In next, we will analyze the external network of Laika through the network and technological terms opened up in the theoretical discussion in sections II and III.

As already stated when discussing the industrial network approach, relationships are characterized by four basic elements: mutuality, interdependence, power relations, and investments made in the relationship (Easton 1992). Mutuality, interdependence, and power relations may vary a great deal from one Open Source project to the next. Dependencies between two projects can be two-way, leading toward mutuality and usually more balanced power relations between the projects.

However, one-way dependencies are also commonplace (i.e., an Open Source project is dependent on another Open Source project but not vice versa). This usually leads to unbalanced power relations between the two projects since only one of the parties of the dyad is dependent on the other.

The structures of projects within the Open Source environment can vary rather a lot in their level of tightness or looseness, as is discussed by Eric Raymond (1999). Within one Open Source project, the position analysis is performed mainly at the level of individuals. But when we leverage the analysis from one project to several, the level of analysis changes to that of entire communities; i.e., we analyze the positions of different Open Source projects against the background of each other.

The level used in network analysis is an interesting issue that has been discussed a great deal by network researchers in general, also outside the Open Source context (see Tikkanen 1998, Möller et al. 2002). In our study, we differentiate between two levels of network analysis, examination within the context of a single Open Source project and consideration involving several Open Source projects and focus on the latter one.

In Table 1, a summary of the analysis of the Laika project in the form of network characteristics is presented.

TABLE 1
NETWORK CHARACTERISTICS OF LAIKA

Network Element	**External network of Laika**
Relationship mutuality and interdependence	Mutual relationship and high interdependency between Maemo and Laika; one-way dependency between Laika and the other projects (e.g., Eclipse)
Relationship investments	Shared goals as drivers of fruitful cooperation – e.g., sometimes priority has been given to the work of another project instead of one's own
Network position and power relations	Laika: critical position as "glue" between other projects but has no power in the other projects
Network structure	Mostly loose networks
Network processes	Evolution – radical when the supercommunity experiences major changes, static otherwise

As already discussed, the Laika project is dependent on many other Open Source communities. Together, Laika, Scratchbox, Maemo, Eclipse, and CDT form a network in which changes in one project create changes in others. The central role of Laika project is to work as glue between the others and integrate them together into a single software tool. Therefore Laika project is even more sensitive for changes in the network structure. Figure 1 illustrates the network formed by communities related to Laika.

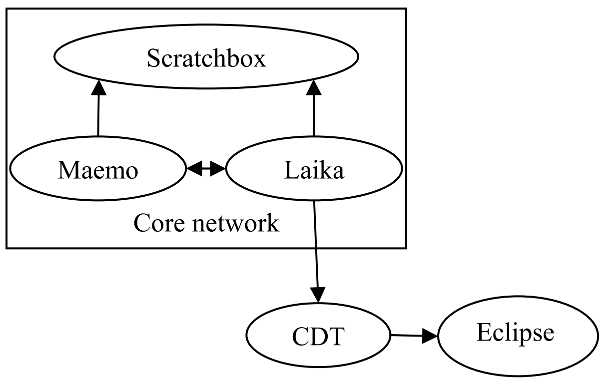

Figure 1 Network formed by communities related to Laika

Laika, Scratchbox, and Maemo form the core of the network whereas CDT and Eclipse are loosely connected to

Laika. The core communities are working towards common goal: make embedded Linux application development easier. Therefore they are closely related and see partner's achievements useful. Between Eclipse and Laika there is only one-way dependency, in which Laika is dependent on Eclipse but not vice versa. In other words Laika get benefit from Eclipse project to a greater extend, but Eclipse does not directly get profit from Laika project.

V. CONCLUSIONS

In this paper, we have discussed industrial network approach as an analytical framework coupled with more technical viewpoints for explaining how Open Source communities work and are linked with each others. As a practical example we used the community developing Laika, an integrated development environment for Maemo, a Linux platform for mobile devices used in, e.g., the Nokia 770 Internet Tablet.

The lessons learned from the experimentation of the community are many. To begin with, it seems obvious that network elements are a key to understanding how communities work and are linked together. In fact, sometimes communities can share responsibilities and create tightly coupled entities that aim at development toward a common goal. In our example, the development of an Open Source mobile device platform benefits from the work of all software developers involved. We can consider this kind of establishment a "supercommunity," or a community of communities that share schedules, goals, and interests. From this perspective, Laika can be seen as a member of the Maemo supercommunity. In industrial network approach terms, it seems in the case of Laika that networks of single communities will broaden into macro networks that have some rather loose network structures but also some very tight ones.

Another interesting discovery is that it is the communities that set their priorities themselves to best benefit the network to which they belong. In the case of the Maemo supercommunity, various communities have sometimes adopted supporting roles to benefit some key community. In exchange, these communities have then received mutual assistance in some other phase of development. This mutuality element has been part of the foci of the industrial network approach literature, and, through the research on OSS communities and networks, we can add new insights to the theoretical debate on networks. In Table 1, a way of summary of the application of network approach to the Laika context in the form of network elements can be seen.

We believe there is much work that we can carry out in the field described in this paper. Below, we provide an outline for future activities concerning Laika, the community maintaining it, and research into the progress of Laika's development.

Concerning Laika, our best prediction is that it will become more and more entangled in the network of Maemo development. Furthermore, while one could assume that actions should be taken to extend the scope of the community to other mobile and embedded Linux environments, we believe that Laika is directly associated with Maemo and that no support is being considered for alternative environments, even if they could benefit from Scratchbox development support. Therefore, assuming that more and more Maemo-based devices are placed on the market, we expect other developers to join Laika, either directly or via plugin technologies that can be integrated into Laika. In a financially oriented environment, such a commitment to a single seminal platform could be considered strategically unwise, which clearly separates community-oriented development from traditional frameworks. At the same time, however, it is conceivable for some development platform other than Eclipse to be supported as well, since this would not alter the mission of the community.

In terms of industrial network approach, we plan to continue monitoring the evolution of the Laika project, as well as the actors participating in the development work. Thus, in our future work, we will concentrate more on the internal network analysis of Laika project. We believe that through in-depth and longitudinal case studies new and fresh insights can be added to the literature addressing internal and external network analysis.

We also wish to study, in the long term, how companies can participate in the development, as well as to observe how funding issues affect the community, potentially leading to the establishment of a company that can take responsibility for some aspects of the community's work, such as helping developers who use the tool. Then, it would be interesting to observe whether the introduction of financial responsibilities changes the manner in which development is organized and how priorities are chosen.

Another direction for further research arises from the industrial network approach perspective. To begin with, we wish to study networks of other communities as well. This will give us a better understanding of how communities are born and evolve, which in turn enables the creation of long-lived Open Source communities fostering growth at other levels. Furthermore, the relationship of communities and companies building on the community contributions is considered an important subject for future study.

REFERENCES

[1] Anderson, J. C., Håkansson, H., & Johanson, J. (1994). Dyadic business relationships within a business network context. *Journal of Marketing,* 58(4): 1–15.

[2] Easton, G. (1992). Industrial networks: A review. In B. Axelsson & G. Easton (Eds.), *Industrial networks: A new view of reality* (pp. 3–34). London: Routledge.

[3] Freeh, V., Madey, G., Tynan R. (2002) - The open source software development phenomenon: an analysis based an social network theory, AMCIS2002

[4] Håkansson, H. & Snehota, I. (1989). No business is an island: The network concept of business strategy. *Scandinavian Journal of Management,* 4(3): 187–200.

[5] Järvensivu, J., Kosola, M., Kuusipalo, M., Reijula, P. and Mikkonen, T. (2006) Developing an Open Source Integrated Development Environment for a Mobile Device. ICSEA'06

[6] Laika (2006). Laika – Scratchbox Eclipse-plugin project. Retrieved from the Web 6 February 2006. http://www.cs.tut.fi/~laika/

[7] Möller, K. & Wilson, D. T. (Eds.) (1995). *Business marketing: An interaction and network perspective*. Kluwer Academic Publishers.

[8] Möller, K, Rajala, A., & Svahn, S. (2002). Strategic business nets – Their types and management. *Journal of Business Research,* 58(9): 1274-1284.

[9] Porter, A., Schmidt, C. (2001) Leveraging open source communities to improve the quality & performance of open source software. In First Workshop on Open-Source Software Engineering, 23rd International Conference on Software Engineering, May 2001

[10] Raymond, E. (1999). *The bazaar and the cathedral*. Sebastopol, CA: O'Reilly.

[11] Tikkanen, H. (1998). The network approach in analyzing international marketing and purchasing operations: A case study of a European SME's focal net 1992–95. *Journal of Business & Industrial Marketing,* 13(2): 109–131.