Master Thesis in Economics

Study of the Evolution of Service Networks

The Case of Salesforce.com AppExchange

서비스 연결망 진화에 관한 연구

: 세일스포스닷컴의 앱거래시장 사례를 중심으로

August 2013

Graduate School of Seoul National University

Technology Management, Economics, and Policy Program

Sodam Baek

Study of the Evolution of Service Networks

The Case of Salesforce.com AppExchange

지도교수 **Jörn Altmann**

이 논문을 경제학석사학위 논문으로 제출함

2013 년 **8** 월

서울대학교 대학원

협동과정 기술경영경제정책 전공

백소담

백소담의 경제학석사학위 논문을 인준함

2013 년 **8** 월

위 원 장 **Junseok Hwang** (인)

부위원장 **Jörn Altmann** (인)

위 원 **Kibae Kim** (인)

Abstract

Study of the Evolution of Service Networks

As IT technology develops, a new business model which motivates consumers to participate in innovation is arising with software-as-a-service (SaaS), which provides utilized software via Internet. Previous studies investigated the structure and evolutionary pattern of networks and found the relationship between network characteristic and innovation performances, but only focused on internal mechanism. Therefore, in this thesis the effort of a platform provider to motivate the participation of third-party developers and users are investigated. Conducting the conceptual background study, the growth, feature of hubs, and the openness of the network was studied with the empirical data gathered from Salesforce.com AppExchange and Developer Market. The results show that platform provider contributed to the growth of service network in the initial periods and that the third-party providers take more roles later periods.

Keywords: Software-as-a-Service, Salesforce.com, Service Networks, Open Innovation, Platform Leader, Collective Intelligence.

Student Number: 2011-23443

Contents

Abstract iii

Contents iv

List of Tables vi

List of Figures vii

Chapter 1. Introduction 1

1.1 Motivation 1

1.2 Problem Description 3

1.3 Research Objective 5

1.4 Research Questions 5

1.5 Research Outline 6

1.6 Contributions 9

Chapter 2. Literature Review 10

2.1 Cloud Computing and SaaS 10

2.2 Service Ecosystem 13

2.3 Open Innovation in Service Networks 16

2.3.1 Social Network Analysis 16

2.3.2 Open Innovation and Collective Intelligence 19

Chapter 3. Methodology 21

3.1 Data Sets 22

3.2 Definition of a Network 23

3.3 Measures of Network Position 24

3.3.1 Centralities 24

3.3.2 Degree Distribution 27

3.3.3 Centralities Map 29

3.4 Openness 30

Chapter 4. Analysis Results 33

4.1 Descriptive Analysis Results 33

4.2 Results of Degree Distribution 35

4.3 Results of Network Position Analysis 38

4.4 Results of Openness Analysis 40

Chapter 5. Conclusion 42

Bibliography 46

Appendix 1: Sample of Application List 53

Appendix 2: Sample of Developer List 54

Appendix 3: Sample of Link list 55

Appendix 4: Applications Network 56

Appendix 5: Sample of Network Position Analysis Results 56

Abstract (Korean) 58

List of Tables

Table 1. Links over Salesforce Lab 41

List of Figures

Figure 1. AppExchange of Salesforce.com...13
Figure 2. The Ecosystem of Salesforce.com..14
Figure 3. Building a Service Network from Salesforce.com AppExchange Data............24
Figure 4. Example of the Service Network of Salesforce.com AppExchange.................27
Figure 5. Betweenness-Degree Centrality Map 30
Figure 6. Growth of the Service Network of Salesforce.com AppExchange 34
Figure 7. The cumulative degree distribution in log-log scales 36
Figure 8. The cumulative degree distribution in log-linear scales 37
Figure 9. Betweenness-Degree Centralities Map...39

Chapter 1. Introduction

1.1 Motivation

As IT technology advances, a new business model which motivates consumers to participate in innovation (e.g. Web 2.0) is arising. Today, software venders provide software-as-a-service, which software users utilize remotely by Internet. The SaaS paradigm appeared and developed with the commercial computing (i.e. punched-card electric accounting machines) in the middle of 1950, but declined with emergence of personal computers. Now, the SaaS paradigm is prominant again with the rise of cloud computing which varies from computing resource service such as Amazon S3 to enterprise management applications such as Salesforce.com (Campbell-Kelly, 2009; Cusumano, 2010).

One of the most important features of today's SaaS implementations is that a SaaS provider attracts the end users as "collective intelligence" into the software innovation (Lévy, 2010; Weiss, 2005). To do so, the SaaS provider provides developers and end users with the marketplace on which they exchange the innovation output. For example, Salesforce.com, a customer relationship management software service, published its platform named as Force.com, in order that users can utilize customized CRM environment in their development of new service (http://www.salesforce.com). Developers can create a new service on their demand with "developer environments, resource, tools, documentation and contents" that Salesforce.com provides by adding the developers' own value (http://developer.force.com). In the service system, the innovation is performed by the developers with their need as well as by the share of innovation resources of SaaS vendors, and the benefit from the innovation is also shared.

Software users and developers interact with each other in the marketplace, and their interaction is represented with a "service network" (Altmann, Meschke, & Mohammed, 2012; Henneberg, Gruber, & Naudé, 2012; Maglio, Srinivasan, Kreulen, & Spohrer, 2006). A service network is defined in a variety of ways, e.g. an extension of value chain from service providers to users (Altmann et al., 2012), and the combination of services for a new service (Kim, Lee, & Altmann, 2013). In this thesis, a service network is defined as a set of nodes and links, which represent the software services created by developers who utilize the marketplace which a SaaS provider opens, and a developer's co-installation of the software services, respectively. The service network characterizes the complementary relationship between the services such as associative support of service to users' requirements.

1.2 Problem Description

The prior research studied the network structure and evolution of the service networks to understand the pattern of innovation by collective intelligence. The prior studies on innovation of service networks (Hwang, Altmann, & Kim, 2009; Kim et al., 2013) follow two research designs. One of the research design investigates the structure and evolutionary pattern of networks (Newman, 2001; Valverde & Solé, 2007; Wagner & Leydesdorff, 2005), and the other design is finding the relationship between network characteristics and innovation performance (Granovetter, 1973; Grewal, Lilien, & Mallapragada, 2006; Krackhardt & Stern, 1988).

In prior researches an assumption is hidden that the service network evolves into a complex structure only by the internal mechanism. For example, preferential attachment for scale-free structure is the heterogeneous link distribution of many real networks (Albert, Jeong, & Barabási, 2000). The model is growing every time step with new nodes which is connected to existing nodes. This model produces a power-law distribution of node degree, which can give the network a small diameter. Another example is small-world structure, which is generated by rewiring the links of a regular network (Watts & Strogatz, 1998). The small world network generated by random rewiring is both highly clusterized and small in diameter.

However, the prior research rarely cares for the effort of the platform provider for promoting the service network evolution by collective intelligence. It is common in real business that a platform provider of the service system strives for an environment in which the system's self-organization by users and developers is successful (Gawer & Cusumano, 2002). For example, Google and Apple both managed to create successful ecosystems around their smartphone operating systems by providing the open application marketplaces which allow third-party developers to create and distribute their applications (Hilkert, Benlian, & Hess, 2011). To govern this ecosystem, they made many efforts to attract third-party developers who regard technical documentation, availability of development-tools, communication with end-users, availability of distribution channel, and technical standards as important features of ecosystem.

Therefore, innovation studies of service networks should consider the effort of platform providers as well as the activity of developers participating in the platform to explain the exact features of the SaaS innovation with collective intelligence.

1.3 Research Objective

The research objective of this thesis is to investigate the effort of a platform provider to motivate the participation of users and developers in the collective innovation within its service ecosystem in the perspective of networks.

1.4 Research Questions

In order to achieve the research objectives, a network analysis was implemented in a case of software innovation system that utilizes collective intelligence, i.e. Salesforce.com. Salesforce.com provides a platform on which users and developers can install and run applications, and a marketplace in which they can exchange the applications. Developers create and modify applications and release the applications by listing them on the marketplace after revision of Salesforce.com. A service network is formed when developers install the application to create new applications. In this network a node is an application released in the marketplace by the developers, and a pair of applications is connected when a developer installs the two applications on its own Salesforce.com platform.

The main question of the research is what a platform provider endeavors for promoting the growth of software innovation ecosystem. To answer this main question, the following three questions are raised about the difference of network position of the platform provider and the other developers releasing their application in the service network: (1) Does the platform provider release a number of applications to motivate its application exchange market? (2) To what network position does the platform provider embed its applications? (3) What role do the applications released by the platform provider take on in the innovation?

1.5 Research Outline

The remainder of this thesis is organized as followings. The next chapter introduces the conceptual background of Salesforce.com, a case of service network. It includes cloud computing and Software-as-a-Service, service ecosystem of Salesforce.com, and open innovation and collective intelligence. Brief information about the CRM software company, Salesforce.com, and its application marketplace, AppExchange, is also demonstrated in this chapter. Moreover, the chapter interprets the basic concepts of social network analysis. It involves real network structure such as small-world and scale-free networks, their evolution model, and indicators measuring network position like centralities.

Chapter 3 demonstrates how to gather the empirical data from the open data source of the study case, AppExchange and Developer Market in Salesforce.com. Also, this chapter describes the process to transform the raw empirical data to network data, and defines the indicators to measure the network position and network structure. Network position of a node is measured with degree centrality and betweenness centrality, and the indicators are used to analyze the network structure through degree distribution and betweenness-degree centralties map. Moreover, Krackhardt and Stern's (1980) E-I index is modified to measure the openness of the platform provider.

Chapter 4 illustrates the analysis results to investigate the research questions according to the indicators defined in Chapter 3. The analysis is performed in three ways: network growth, the features of hubs and the openness of the platform provider. The result of network growth says that Salesforce.com released more applications than the other developers in the initial periods. However, its share of release gradually decreased, so the other developers dominated the application market in the last periods. The results on the features of hubs show that the service network has four hubs, and the hubs are released by the third party developers. Finally, the result about the openness of the platform provider reveals that the applications released by Salesforce.com are connected more with the applications released by the third party developers than the other applications of Salesforce.com.

The analysis results suggest that the service network was led by the platform provider in the initial periods and the third party developers gradually took more roles in software innovation on the platform, and the service network is currently evolving through the vigorous participation of third party developers as collective intelligence and their cooperation with the platform provider. This conclusion is summarized in Chapter 5.

1.6 Contributions

The conclusion of this research raising the importance of a platform provider's effort in the initial periods of operating the platform gives both academic and practical implications. The conclusion of this research orients the interest of innovation studies of service networks into the role of a platform provider in the innovation through collective intelligence. The prior research emphasized that service networks evolve into complex structure (e.g. scale-free and small-world topologies) by user developers (Albert et al., 2000; A.-L. Barabási & Albert, 1999), and that the network position in the complex network is related with innovation performance (Tsai, 2001). However, the prior research misses the role of a platform provider in the service network. The results of this research reveal that a platform provider strives for motivating the evolution of a service network especially in the initial periods. The results raise several issues concerning innovation in service networks, including the effect of the platform provider's effort on the success of the platform, and the conclusion provides platform leaders with the necessity for endeavoring to motivate the users, or the user-developers, to participate in the innovation on its platform.

Chapter 2. Literature Review

2.1 Cloud Computing and SaaS

Cloud computing is a service that customers can rent storage, compute, and use network capacity from a company that provides its resources using its data center and making it available via the Internet (Smith, 2009). Cloud computing is a new computing paradigm provided as a service in which consumers are able to customize computing infrastructure over the network (Youseff, Butrico, & Da Silva, 2008). In cloud computing, the ability of service-composition enable service providers to aggregate existing services and create new services by allowing customized solutions and changing distribution models (Leimeister, Böhm, Riedl, & Krcmar, 2010).

The data center for hardware and software is called a cloud and it can be divided in three types. If a cloud is paid for, an amount of usage to the general public from an off-site third party provider who shares resources via the Internet, it is called a public cloud (Armbrust et al., 2010; Rimal, Choi, & Lumb, 2009). When a cloud is made of internal data centers of a company or an organization without the restrictions of network bandwidth, security exposures and legal requirements, and not available to the general public, it is called private cloud. A cloud could be hybrid when the cloud is mixed with the public and private clouds.

Information Technology (IT) software is one of the necessary goods for modern enterprises, but the high cost for IT innovation is a barrier to startup companies (Qu & Ye, 2010). Cloud computing is a solution to eliminate this barrier by changing the computing product to service that companies can purchase and utilize it on their own demand. The cloud computing model provides new opportunities to consolidate individual component services to create value-added, complex services based on cloud computing platforms provided by platform providers and new roles can be found in cloud computing.

When something is delivered to the end user as a Service, it is called X-as-a-Service (XaaS), and X can be software, hardware, platform, infrastructure, data, business, desktop, framework, organization etc (Rimal et al., 2009). The most popular three XaaS among these are Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). SaaS is a multi-tenant platform which commonly provides applications based on common resources and database. Examples of SaaS are Salesforce.com, NetSuite, Oracle, IBM, and Microsoft etc. PaaS provides innovators with all the systems and environments which are the womb for innovators in which they can develop, test, deploy and host applications on clouds. A service of computer infrastructure on cloud is called IaaS which is beneficial because of usage-based payment scheme and lastest technology. Examples are GoGrid, Flexiscale, Layered Technologies, Joyent and Mosso/Rackspace etc.

Salesforce.com is one of the biggest Customer relationship management (CRM) companies which provide their service as SaaS. Salesforce.com is a platform provider and application provider which hosts third party developers to participate in its infrastructure (Qu & Ye, 2010). Third party developers create and modify their applications, or the environment for innovation, on the platform of Salesforce.com by collaborating with the other developers. Users can customize and utilize a number of applications on this cloud based platform to fit their needs of CRM solution.

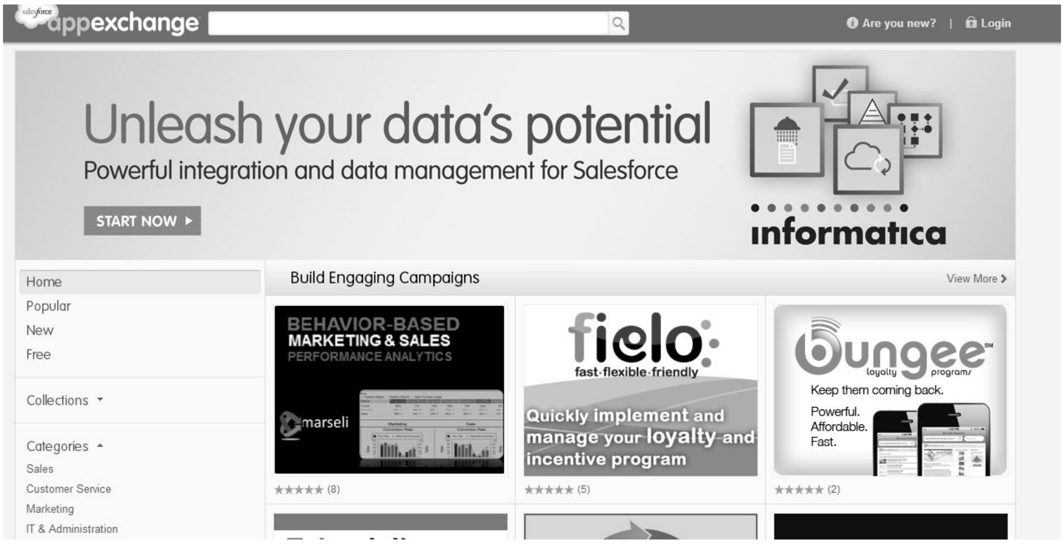


Figure 1. AppExchange of Salesforce.com

2.2 Service Ecosystem

A Web service ecosystem is a logical collection of Web services in which customers consume services through variable distribution and delivery channels (Barros & Dumas, 2006). In web service ecosystems, a measure of its success is how well a service is connected with others. The first beneficiaries of open marketplaces of web service are innovative companies such as Salesforce.com, StrikeIron and Grand Central. These companies integrated enterprise application developed by the other software developers using XML and Web service technology.

Figure 2 describes the innovation ecosystem of Salesforce.com. The ecosystem consists of all involved services and solution suppliers such as platform provider, application provider and end users. Developers can code with testing and deployment tools with which Salesforce.com provides. Salesforce.com can absorb the most fitting software components from third party developers which encourages further growth through integration with other components.

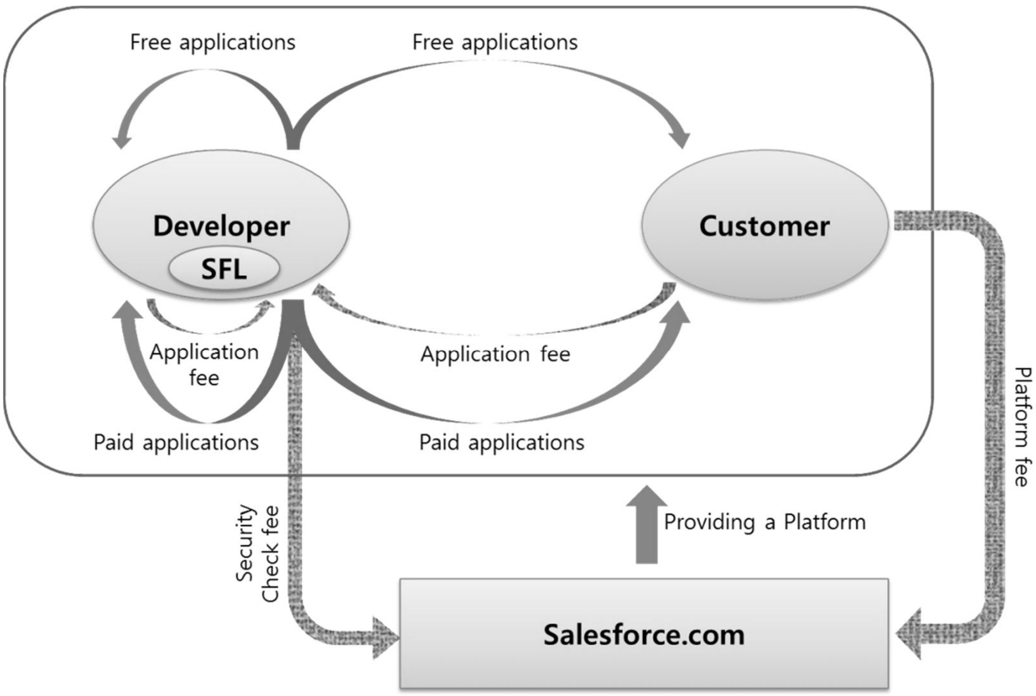


Figure 2. The Ecosystem of Salesforce.com

In Salesforce.com AppExchange market, Salesforce.com provides the platform to both customers and developers. Customers pay Salesforce.com for using the platform and developers can use the platform for free. However, the developer edition and customer edition of the platform have some differences. The developer edition has limitations in its data storage, file storage, API request number, number of application, and bandwidth (wiki.developerforce.com), while the customer edition does not. Developers develop application based on this platform, and they decide whether to release this application free or charged. Before they list their application on the marketplace, the application should pass the security check which Salesforce.com provides. If they publish the application free of charge, Saleforce.com checks the security for free. However, if the developer wants to sell the application, they need to pay $5,000 for the first security check and $2,500 annually to keep their application listed on AppExchange, the marketplace of applications created on the Salesforce.com platform. Developers can be users of the applications created by the other developers, and reuse and combine the applications for their innovation. In this case, they also pay for the charged applications to other developer who developed the application. Among these developers, there is Salesforce Labs which is the developer's account of Salesforce.com. It develops and releases the applications to motivate the participation of third party developers.

2.3 Open Innovation in Service Networks

2.3.1 Social Network Analysis

Modern social network studies started with the questions what real networks look like and how the networks are formed. One of the pioneers is the experiment on the US citizens to measure their social networks (Milgram, 1967). In the experiment of Milgram (1967), he sent letters reading the destination to 500 randomly chosen people living in the United States of America and asked them to transmit the letters to their friends who are most likely to know the destination. The experiment result shows that the letters are received through only five to six intermediaries in average. Six links between a certain pair of US citizens is very small comparing to its population; there were about 200,000 people in the US in 1960s.

By the late 1990s, some statistical physicists investigated the structure of real large networks and their evolution with the favor of advanced computation power. They found that the real networks are not homogeneous according to the assumption of network studies that networks are almost homogeneous (Albert, Jeong, & Barabási, 1999). The hyperlink network of World-Wide Web, for example, is scale-free and there are few hubs connecting a majority of nodes which has only few links, and these hubs make the network small world (Albert et al., 1999). Barabási and Albert (1999) proposed a preferential attachment model, in which network grows and a new node prefers the node with the large number of links. Watts and Strogatz (1998) proposed a random rewiring model in which the self-organization from a regular network by random rewiring produces a small world network, in which nodes are highly clustered but the distance among them are not so large in average.

The inhomogeneous networks whose degree distribution decays by a power function and dominated by a few hubs are called scale-free network (Albert et al., 1999). The scale-free topology appears in a variety of real networks including academic collaboration networks (A. L. Barabási et al., 2002) and Web service innovation networks (Hwang et al., 2009). Scale-free networks are surprisingly tolerable against random failures, and the diameter of these networks increases rapidly (Albert et al., 2000) .

Innovation studies considered that the network position could affect the innovation performance if the network structure is inhomogeneous (C. Freeman, 1991). One of the most popular techniques of measuring network position is using centralities: Degree, betweenness centralities, and closeness centralities (L. C. Freeman, 1979). These indicators measure how deep a node is embedded in a network. Degree centrality is the number of links that a node has with its neighbors. Betweenness centrality is based upon the frequency with which a vertex falls between pairs of other vertices on the shortest path or geodesic linking them. Closeness centrality is the inverse of the sum of distance of a node to the other nodes in a network.

The network position of a node is identified with the centralities. For example, in the prior research, Everard and Henry used degree centrality and betweenness centrality to examine a pattern of interlocked directorates among the top 50 e-commerce firms and analyzed how this network is different from the network of other dominant firms (Everard & Henry, 2002). Granovetter (1973) called the links connecting the nodes which has low degree, but high betweenness centralities which has weak ties, and proposed that the weak ties play an important role in innovation. Kim, Cho & Kim (2012) used a map spanned by degree and betweenness centralities to investigate the systematic pattern of innovation strategy. The effect of network position is also dependent on the context of innovation. The central position gives high performance if the innovator has good capacity for absorbing the innovation resources (Grewal et al., 2006; Tsai, 2001).

2.3.2 Open Innovation and Collective Intelligence

Open innovation is a new trend of innovation introduced in early 2000s. In open innovation, an innovating company invites the third party innovators in its innovation to aggregate the innovation resources out of itself (Chesbrough, 2003). This is an economic way comparing to the old style of innovation in which a company must possess all the innovation resources on its innovation process from R&D and commercialization. The leading companies in Information Technology (IT) gained their position with the open innovation strategy. For example, Intel focused on developing and upgrading Central Processing Unit (CPU) and opened its interface in order that the third party innovators access to the core technology. In this way, Intel grew in the innovation "ecosystem" in which its partners utilize Intel CPU as a platform and the platform's value increases as the partners develop complementary goods (Gawer & Cusumano, 2002).

In the open innovation environment, the success of the innovation relies on the participation of the third party developers in the innovation on the basis of platform that a leading company provides. That is, the "platform leader" provides its platform on which the third party developers can access and reuse the innovation resources and collaborate with each other (Gawer & Cusumano, 2002). Further, the platform leader utilizes its partners as "collective intelligence" to achieve the innovation with low cost in efficient and effective way (Lévy, 2010; O'reilly, 2007). Then the platform leader gains benefit from the commercial side while it gives up benefit in the side for collective intelligence (Cusumano, 2010). Gloor et al. (2009) called utilizing the third party as collective intelligence "Wisdom of Crowds." Also, they introduced a new set of social network analysis based algorithms for analyze the Web and social networks by visualizing the social networks, semantic mining, and analyzing the text of the social networks (Gloor, Krauss, Nann, Fischbach, & Schoder, 2009).

Chapter 3. Methodology

The service network describing the ecosystem that Salesforce.com provides, which was described in Chapter 2, is analyzed with several social network analysis indicators. The analysis is accomplished with the following three steps: First, the information of Salesforce.com applications and developers including which application they installed on its platform was gathered from the open platform for exchanging applications that Salesforce.com provides. Second, a service network was defined as a set of applications and their co-installation, and the definition was applied to the gathered data. Finally, the service network was analyzed with two indicators measuring network position (i.e. degree and betweenness centralities) and one index describing the openness of the platform provider to the other application developers. The focus of the network position analysis is on identifying the characteristics of network position of applications released by Salesforce.com, distinguished from the one of the third party developers. The openness of the platform provider was measured with a modified version of E-I index designed by Krackhardt and Stern (1980).

3.1 Data Sets

The empirical data of the analysis was gathered from the open platform of Salesforce.com. The information of application specification was gathered from the marketplace of Salesforce.com, AppExchange (http://appexchange.salesforce.com). AppExchange is a marketplace of applications of Salesforce.com that customers can purchase and download applications they need and developers can upload and sell their applications under the control of Salesforce.com. The information includes the list of applications and each application contains information of provider, released date, pricing, categories, reviews and rating. Among them the information used in the analysis is application name, the provider of the application and its released date. A sample of the gathered data is shown in Appendix 1.

The information of application co-installation by a user-developer was gathered from the platform for developers, provided by Salesforce.com (http://appexchange.salesforce.com/developers). The data involves developers' name (i.e. their Salesforce.com account) and the applications each developer installed in its system. A sample of developers list is described in Appendix 2.

3.2 Definition of a Network

In this thesis, a service network is defined as a set of nodes and links, which representing the applications operating on the Salesforce.com platform and the co-installation of the applications in a developers system, respectively. According to the empirical data set gathered from AppExchange, each node of the service network is identified by its name, and it involves two attributes, or the application developer and its released date (Appendix 1).

In this thesis, it is defined that a pair of nodes is linked when a developer install both of the nodes in its system. Figure 3 shows an example of building links between applications. On the left side of Figure 3, there are four applications (i.e. App 1, App 2, App 3 and App 4) and two developers (i.e. Dev A and Dev B). Dev A installed App 1 and App 2, and Dev B did App 1, App 2, App 3 and App 4. By definition, App 1 and App 2 are connected because Dev A installed the two applications. And, App 2, App 3 and App 4 are connected with each other because Dev B installed the three applications. Therefore, a service network is formed like the right side of Figure 3. A sample of network links between applications is described in Appendix 3 and the application network is visualized with UCINET in Appendix 4.

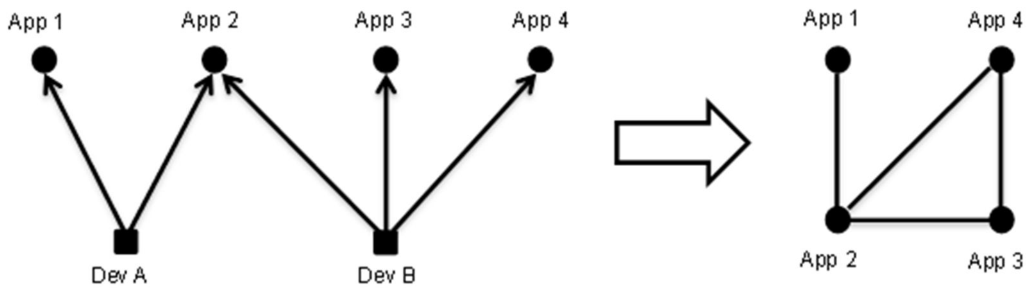


Figure 3. Building a Service Network from Salesforce.com AppExchange Data

3.3 Measures of Network Position

3.3.1 Centralities

Centralities indicate the network position of nodes. There are a variety of centralities including degree and betweenness centralities (L. C. Freeman, 1979). The degree centrality of a node measures the number of nodes connected with the node. It is most intuitive and simple way to measure how deep the node is embedded in its network. According to Nieminen's (1974) notation, the degree centrality $C_d\ (p_k)$ of the k-th node p_k in a network with size n is defined as:

$$C_D(p_k) = \sum_{i=1}^{n} a(p_i, p_k) \cdots \text{Eq. (1)}$$

$a(\cdot, \cdot)$ represents the link between two nodes p_i and p_k. $a(p_i, p_k) = 1$ if and only if p_i and p_k are connected with each other, $a(p_i, p_k) = 0$ if and only if p_i and p_k are not connected. $C_D(P_k)$ is large if node p_k is directly connected with a large number of other points. $C_D(P_k) = 0$ if a node is totally isolated from contact with the other nodes.

Betweenness centrality, which is based on the number of geodesics, is also a popular indicator for measuring network position (L. C. Freeman, 1979). It is the number of shortest paths passing through a certain node among all possible shortest paths, and implies how much the node mediates the other nodes. Similar to the definition of degree centrality, the betweenness centrality $C_B(p_k)$ of node p_k is defined as

$$C_B(p_k) = \sum_{i,j<i}^{n} b_{ij}(p_k) \cdots \text{Eq. (2)}$$

Here, $b_{ij}(p_k)$ is the ratio of the number $g_{ij}(p_k)$ of geodesics between any combination of nodes p_i and p_j passing through node p_k to the total number g_{ij} of geodesics between any nodes p_i and p_j.

$$b_{ij}(p_k) = \frac{g_{ij}(p_k)}{g_{ij}} \cdots\cdots\cdots \text{Eq. (3)}$$

By definition, C_B (p_k) is 1 if all the geodesics pass through node p_k and 0 if there is no geodesic traversing node p_k.

Figure 4 is a sample of the service network of Salesforce.com consisting of seven nodes, four nodes (i.e. p_1, p_2, p_3 and p_4) of which are released by Salesforce.com, and the remainder (i.e. p_5, p_6 and p_7) are released by the third party developers. In this graph, for example, the degree and betweenness centralities of p_4 and p_6 are C_D (p_4) = 3, C_D (p_6) = 2, C_B (p_4) = 4.5 and C_B (p_6) = 1. A sample of a degree and betweenness centralities of co-installation network of Salesforce.com AppExchange is described in Appendix 5.

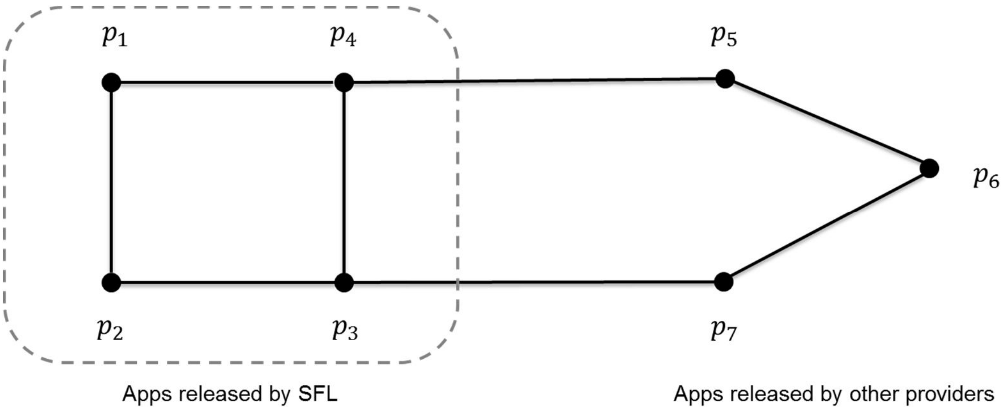


Figure 4. Example of the Service Network of Salesforce.com AppExchange

3.3.2 Degree Distribution

The distribution of degree centralities in a network is used to measure the topology of the network, while the degree centrality of a node is used to identify the network position of the node. The degree distribution is represented by the frequency $P(C_D)$ of a certain degree centrality value C_D. If a network is regular, the degree distribution is $P(C_D=\hat{C}) = n$ and $P(C_D \neq \hat{C}) = 0$ when each node is directly connected with $\hat{C}$ nodes. If a network is random, the degree distribution is like Gaussian and the network topology can be characterized by the average of the degree centralities.

According to empirical network analysis in prior research, the degree distribution of many large, complex networks is scale-free (Albert et al., 1999). The frequency of degree in scale-free networks is represented by a power function: $P(C_D) \sim C_D^{-\gamma}$. The degree distribution in log-log scale decays by a negative linear function:

$$\log P(C_D) \sim -\gamma \log C_D \quad \text{Eq. (4)}$$

Because the noise is considerable in the high degree centrality area due to the low frequency in that area of a scale-free network, the cumulative degree distribution is proposed to analyze the scale-free topology (Newman, 2005). The cumulative degree distribution is the frequency of the degree centralities larger than a certain value. The formula of cumulative degree distribution in log-log scale is equivalent to that of degree distribution because the integration of a power function is also a power function:

$$\log P(C_{D>}) \sim -(\gamma + 1) \log C_D \quad \text{Eq. (5)}$$

If the degree distribution is exponential, i.e. $P(C_D) \sim \exp(-\beta C_D)$, the degree distribution in log-lin scale decays by a negative linear function: $\log P(C_D) \sim -\gamma C_D$. And, the integration of the exponential degree distribution reduces the noise at the high degree centrality area and does not change the formula like the scale-free network:

$$\log P(C_{D>}) \sim -\beta C_D \quad \text{Eq. (6)}$$

3.3.3 Centralities Map

Centrality map is the space scaled by betweenness and degree centralities, in which a node locates according to its betweenness and degree centralities (Figure 5). The centrality map is used to identify the network position of nodes. The network position of node is classified into four categories: hubs, cores and bridges. First, a node with both high degree and betweenness centralities is a hub, which connects the entire network in short distance (Albert et al., 1999). Second, a node is a bridge if its betweenness centrality is high but its degree centrality is low (Everard & Henry, 2002). Third, a node with high degree centrality but low betweenness centrality is a core. A core could locate at the center of a cluster, but it does not connect the entire network.

The criterion dividing the categories is flexible. If the degree and betweenaness distributions of a network are Gaussian, the criterion for the classification could be the median, average and the 4^{th} quartile can be used. If the distribution is scale-free, on the other hand, the nodes with largest degree and betweenness centralities could be selected.

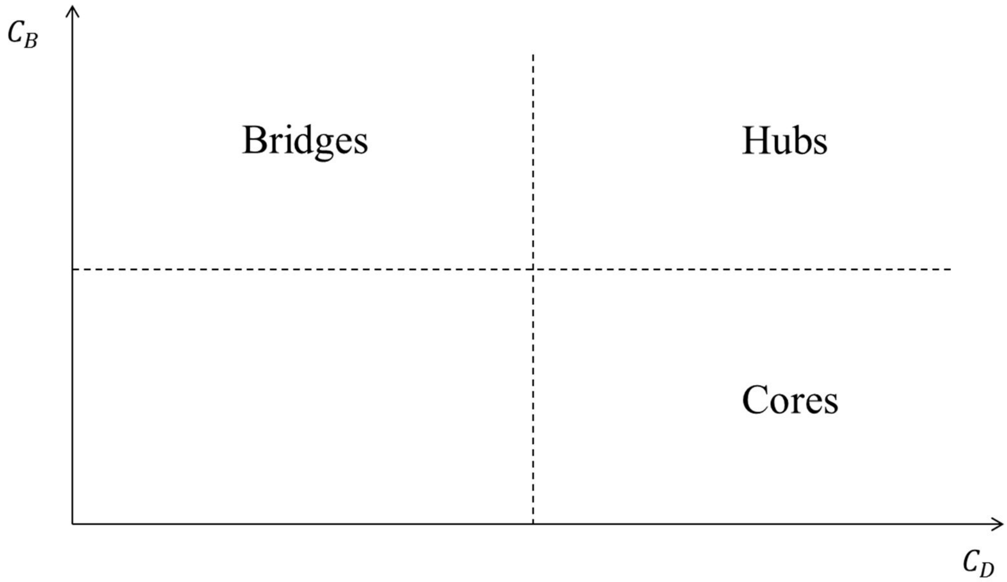


Figure 5. Betweenness-Degree Centrality Map

3.4 Openness

A real network usually consists of nodes whose attributes are inhomogeneous. The nodes which have the same node attribute is called a subgroup, or a group in short (Kibae, 2011). Krackhardt and Stern (1988) developed a simple index, what is called E-I index, measuring the openness of the subgroups in a network. A link between any two nodes in a network is distinguished into internal and external links. Internal links *IL* represent the links between the nodes belonging to a same subgroup, and external links *EL* the links between the nodes belonging to different subgroups. With the internal and external links Krackhardt and Stern's E-I index is defined as:

$$EI = \frac{EL-IL}{EL+IL} \quad \text{Eq. (7)}$$

The index score ranges from -1.0 to +1.0, and increases as there are more external links in a network. The index score is -1.0 if all the links in a network are internal. On the other hand, the score is +1.0 if all the links are external. The score is 0 if the number of internal links is equivalent with the number of external links.

The E-I index is modified to analyze the openness of one subgroup to the other subgroups, and this modified E-I index is called the group E-I index (Everett & Borgatti, 2012). In this modification, internal links are the links among the nodes involved in a specific subgroup, and external links the links connecting the nodes in the subgroup with the nodes out of the subgroup. In Figure 4, for example, internal links are the links (p_1,

p_2), (p_2, p_3), (p_3, p_4) and (p_1, p_4) connecting the nodes p_1, p_2, p_3 and p_4 belonging to Salesforce.com category. And external links are the links (p_4, p_5) and (p_3, p_7) connecting the nodes in Salesforce.com category and the nodes out of the category. With the internal links IL_i and external links EL_i for a chosen group i, the group E-I index EI_i for group i is defined as:

$$EI_i = \frac{EL_i - IL_i}{EL_i + IL_i} \quad \text{Eq. (8)}$$

Chapter 4. Analysis Results

4.1 Descriptive Analysis Results

According to the data of application co-installation data gathered from Developers' platform, 74 developers published their installation information among 1000 developers registering in the developers' platform. The 74 developers utilized 205 applications which are released by the other developers to create new applications. The number of developers of the 205 applications is 99.

The application list gathered from AppExchange describes the application specification of the 205 applications, including the provider, the pricing policy, service categories and released date. During the study period, from July 2005 to April 2013, the number of released applications increased ceaselessly. Among the 205 applications, 74 applictions are provided by Salesforce Lab, the developer account of Salesforce.com. However, the contribution of Salesforce.com releasing applications decreases gradually, while its contribution was higher than the initial periods.

Figure 6 shows the trend of the contribution of Salesforce Lab and the third party developers to releasing applications. The number of applications released by Salesforce

Labs (SFL) steadily increased from 9 in January 2006 to 28 in January 2008. On the other hand, the increase of the number of applications released by the third party developers was tardy in these periods. The number of applications released by the third party developers was 4 and 16 in January 2006 and January 2008, respectively. However, the release by the third party developers boosted around January 2009 and their contribution surpassed the number of applications released by Salesforce Lab in April 2010. The numbers of applications of the third party developers were 24 in January 2009 and 127 in February 2013 and those of Salesforc Lab were 36 and 74 in the same periods.

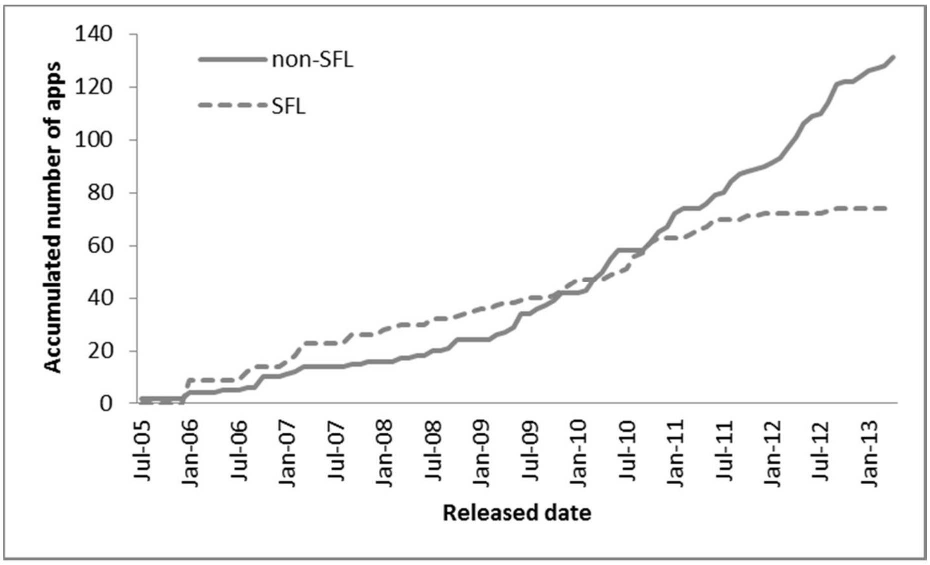


Figure 6. Growth of the Service Network of Salesforce.com AppExchange

4.2 Results of Degree Distribution

The analysis results of cumulative degree distribution are described in Figure 7 and Figure 8. Like the degree distribution analysis achieved in the prior research (Albert et al., 1999), Figure 7 describes the relation between the degree centrality score and the number of nodes not less than score in log-log scales. The result shows that the cumulative degree distribution seems to decay by a linear function in log-log scales. The ordinary least square (OLS) regression for two variables was applied to testing if the cumulative degree distribution in log-log scales fits to a linear function. The slope of the function is -1.0412, the t-test statistic of slope is 0.000, and the adjusted R^2 is about 0.799. The results say that the correlation between the two variables is very significant. The explanation power, or adjusted R^2, is large comparing to the statistical analysis in social science, but is small comparing to those in natural science.

The analysis results of cumulative degree distribution in Figure 7 suggest that the service network could be scale-free. However, the graph shows an extraordinary feature; the degree distribution is concave below. It implies that the cumulative degree distribution could decay by the other function instead of a power function. Therefore, the cumulative degree distribution is drawn again in log-linear scales.

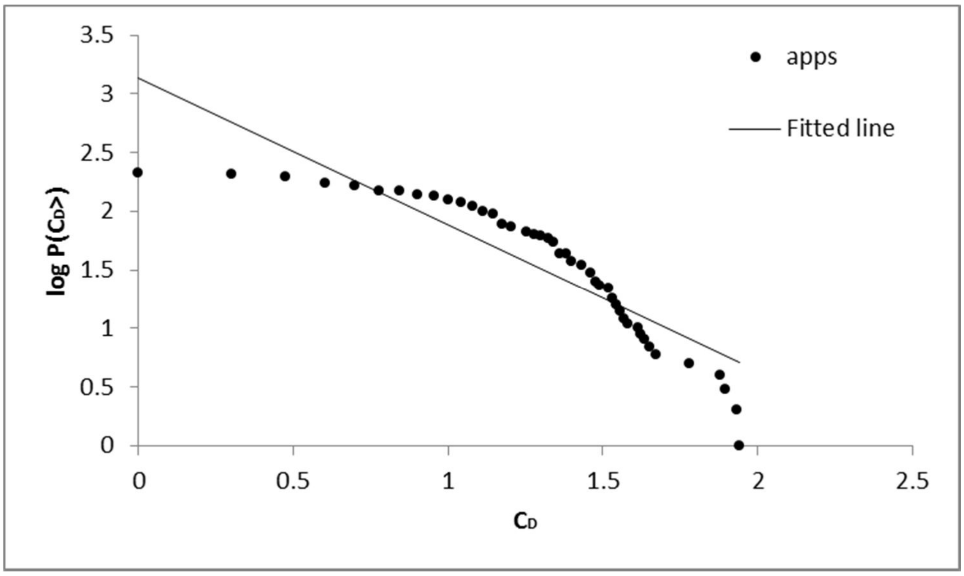


Figure 7. The cumulative degree distribution in log-log scales

Figure 8 depicts the relation between the degree centrality score and the number of nodes with degree centrality equal to or larger than the score in log-linear scales. The cumulative degree distribution in the figure looks decaying by a linear function except the five outliers whose degree centrality is larger than 55. The OLS regression of the cumulative degree distribution in log-linear scales except the five outliers produces the slope, the t-test statistic of slope and the adjust R^2 0.0347, 0.000 and 0.993, respectively.

The results imply that the degree distribution of the service network fits to an exponential function with higher significance and explanation power than the power-law model if the five outliers are ignored.

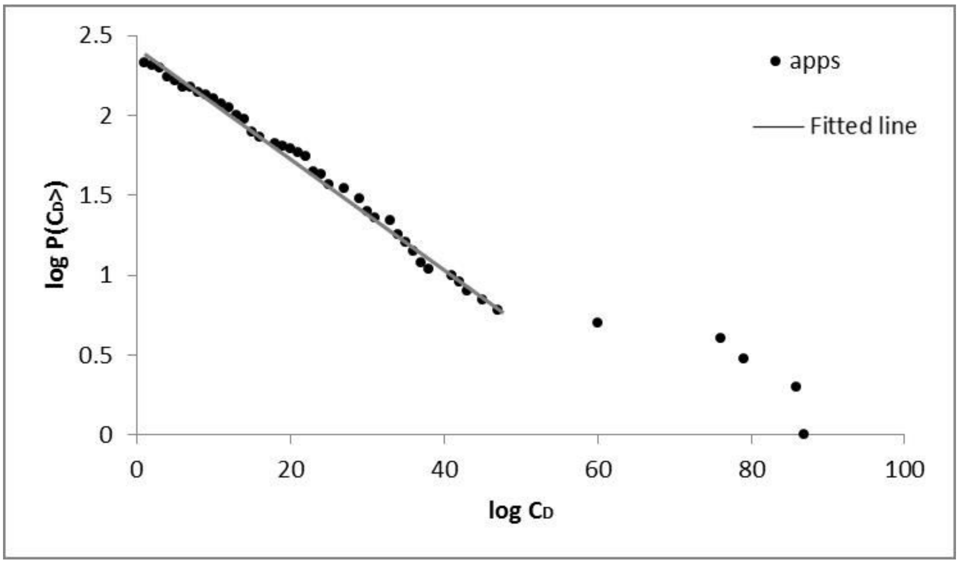


Figure 8. The cumulative degree distribution in log-linear scales

The outliers are "Inside View Free," "Conga Composer," "Mass Update And Mass Edit From List View," "Appirio Cloud Sync for Google Apps," and "Field trip." The attributes of these services are various. First, the service categories of these applications are Sales Intelligence, Document Generation, Admin & Developer Tools, Email & Calendar Sync, and Admin & Developer tools in order. Second, the developers releasing these applications are InsideView, Conga, Salesforce Lab, Appirio and Qandor, respectively. Finally, their released dates are August 2006, November 2009, June 2009, November 2011 and November 2010. Therefore, it is hard to identify the factors determining the central nodes in the service network.

4.3 Results of Network Position Analysis

Figure 9 depicts the nodes in the service network of Salesforce.com AppExchange locating on the map spanned by betweenness and degree centralities. The five outliers in *Figure 8* are clearly deviated from the other nodes in the betweenness-degree centralities map. The network positions are divided by the two lines at $C_B = 1600$ and $C_D = 55$. Four of the five outliers occupy the hub position, both degree and betweenness centralities of which are higher than those of the other nodes. They are "Inside View Free," "Conga Composer," "Appirio Cloud Sync for Google Apps," and "Field trip." The remainder of the outliers is a core, which has high degree centrality but low betweenness centrality. It is the node representing "Mass Update And Mass Edit From List View" released by Salesforce Lab. The nodes except the outliers locate in the position with low degree and betwenness centralities, and there is no node in the bridge position. The applications released by Salesforce Lab do not show significant difference in their position from those released by the third party developers.

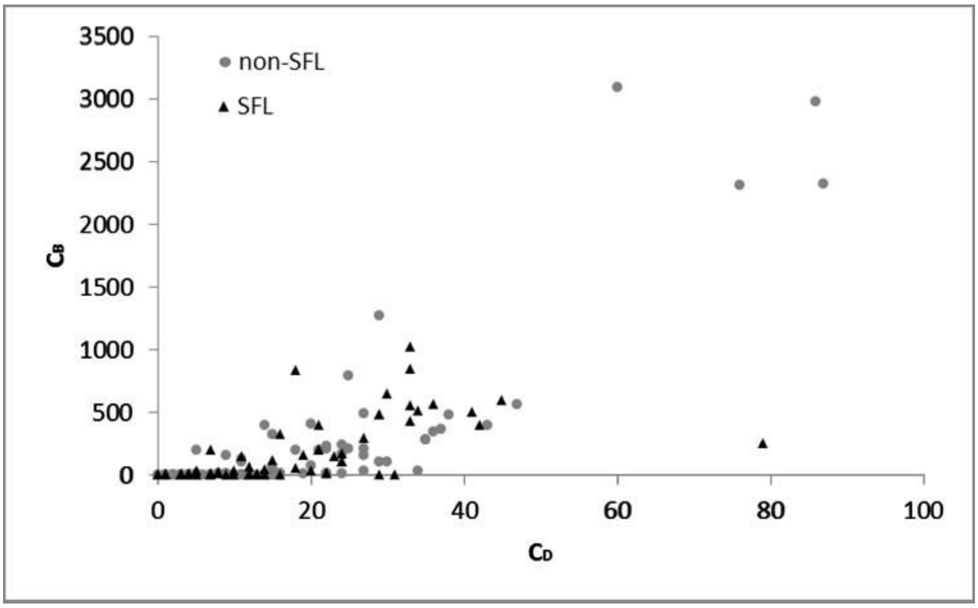


Figure 9. Betweenness-Degree Centralities Map

The results show whose applications dominate the service network. The applications locating at the hub position are what were released by the third party developers. And, a majority of applications are released by the Salesforce Lab, the platform provider. That is, the service network is led by the third party developers while the platform provider released a lot of applications to promote the network to be active.

This is desirable according to the design of a service ecosystem depicted in *Figure 2*, in which the platform provider invites the user-developers in the innovation as collective intelligence. Providing the innovation environment of the platform provider by releasing the applications was success to attract the developers to participate in the innovation.

4.4 Results of Openness Analysis

To analyze the openness of the platform provider, the group E-I index was calculated. Table 1 is the summary of the internal links within and the external links over Salesforce Lab. The service network consists of 2247 links as a whole. Among them, 397 links connect the applications both released by Salesforce Lab, and 1060 links connect the applications one of which is released by Salesforce Lab and the other of which by the third party developer.

According to the definition of group E-I index defined in Section 3.4 and the number of links described in Table 1, the group E-I index EI_{SFL} for Salesforce Lab is calculated as 0.455.

$$EI_{SFL} = \frac{EL_{SFL} - IL_{SFL}}{EL_{SFL} + IL_{SFL}} = \frac{1060 - 397}{1060 + 397} = 0.455 \quad \text{Eq. (9)}$$

The group E-I index score is positive and considerably high comparing to the range of the group E-I index, -1.0 to +1.0. The result suggest that the applications released by Salesforce Lab are connected with roughly three times as many applications released by the third party developers as the applications released by Salesforce Lab. That is, Salesforce Lab is considerably open to the third party developers.

Table 1. Links over Salesforce Lab

Type of links	Number of links
All the links	2247
External links over Salesforce Lab (EL_{SFL})	1060
Internal links within Salesforce Lab, (IL_{SFL})	397

Chapter 5.   Conclusion

5.1 Outlook and implications

In this thesis, network analysis was achieved for the service network of applications linked by their co-installation relation, gathered from Salesforce.com AppExchange and Developers' platform. The analysis results suggest that the effort of platform provider for attracting the third party developers takes an important role in the evolution of its service network. Salesforce Lab, the developer account of the platform provider, released about one third of applications on its platform. Many of applications released by Salesforce Lab were linked with the applications released by the third party. It means that the Salesforce Lab's applications are complementary to the applications of the third party, so that they increase the utility of the applications of the third party developers. Finally, the applications released by the third party developers dominate the service network instead of the platform providers. In the results, four hubs are the applications released by the third party. This implies that the platform provider does not gain the performance in the service network though it contributes much to the network evolution.

The results give both academic and practical implications. In academic perspective, the findings in the thesis orient the service network studies into the effort of platform provider to motivate the participation of the third party developers. The prior research of innovation in network approach emphasizes that service networks evolve into complex structure by the third party's participation in innovation as collective intelligence. Further, it is basically assumed that the network evolution is governed by internal mechanism such as preferential attachment and random rewiring (A.-L. Barabási & Albert, 1999; Watts & Strogatz, 1998). In this assumption, a platform provider does not contribute the network evolution except it provides with the platform and manages it to work properly. However, the results of empirical analysis in the thesis show that the platform provider also contribute to the network evolution as well as the platform management, and the contribution motivates the third party developers to participate in the innovation on its platform.

The academic implication above is also practical for platform providers in managerial perspective. In the current models of software innovation using collective intelligence, the roles of a platform provider is separated from the role of the third party developers. According to the model, the third party developers share their innovation resources, reuse and recombine them for innovation, while the platform provider prepare the ecosystem in which the developers work. However, the platform provider's preparing good ecosystem, e.g. the ecosystem according to the open innovation allowing the third party's access to core technology of platform provider shown in Gawer and Cusumano (2002) does not lead certainly to the success of innovation by collective intelligence. The empirical analysis results of the thesis show that there is the participation of platform provider in the innovation in early periods in order to motivate the third party developers.

5.2 Limitations

This thesis re-orients the interest of innovation studies on service networks evolving by collective intelligence to the effort of platform provider. However, it has several limitations due to the concentration only on finding the effort of platform provider in the service network evolution. Most of all, the findings require more rigorous analysis in detail. The results propose that there are hubs, but the thesis does not give common factors giving the nodes the hub position. It is also important to validate if it is common that a platform provider contributes directly to the innovation on its platform and if its desirable contribution is related with the success of innovation on its platform. To resolve these limitations, network analysis should be implemented in more cases.

Bibliography

Albert, Réka, Jeong, Hawoong, & Barabási, Albert-László. (1999). Internet: Diameter of the world-wide web. *Nature, 401*(6749), 130-131.

Albert, Réka, Jeong, Hawoong, & Barabási, Albert-László. (2000). Error and attack tolerance of complex networks. *Nature, 406*(6794), 378-382.

Altmann, Jörn, Meschke, Martina, & Mohammed, Ashraf Bany. (2012). A Classification Scheme for Characterizing Service Networks: Seoul National University; Technology Management, Economics, and Policy Program (TEMEP).

Armbrust, Michael, Fox, Armando, Griffith, Rean, Joseph, Anthony D, Katz, Randy, Konwinski, Andy, . . . Stoica, Ion. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50-58.

Barabási, A. L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., & Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications, 311*(3–4), 590-614. doi: http://dx.doi.org/10.1016/S0378-4371(02)00736-7

Barabási, Albert-László, & Albert, Réka. (1999). Emergence of scaling in random networks. *science, 286*(5439), 509-512.

Barros, Alistair P, & Dumas, Marlon. (2006). The rise of web service ecosystems. *IT professional, 8*(5), 31-37.

Campbell-Kelly, Martin. (2009). Historical reflections The rise, fall, and resurrection of software as a service. *Communications of the ACM, 52*(5), 28-30.

Chesbrough, Henry William. (2003). *Open innovation: The new imperative for creating and profiting from technology*: Harvard Business Press.

Cusumano, Michael. (2010). Cloud computing and SaaS as new computing platforms. *Communications of the ACM, 53*(4), 27-29.

Everard, Andrea, & Henry, Raymond. (2002). A social network analysis of interlocked directorates in electronic commerce firms. *Electronic Commerce Research and Applications, 1*(2), 225-234.

Everett, Martin G., & Borgatti, Stephen P. (2012). Categorical attribute based centrality: E–I and G–F centrality. *Social Networks, 34*(4), 562-569. doi: http://dx.doi.org/10.1016/j.socnet.2012.06.002

Freeman, Christopher. (1991). Networks of innovators: a synthesis of research issues.

Research policy, 20(5), 499-514.

Freeman, Linton C. (1979). Centrality in social networks conceptual clarification. *Social networks, 1*(3), 215-239.

Gawer, Annabelle, & Cusumano, Michael A. (2002). Platform leadership: how Intel. *Microsoft, and Cisco Drive Industry Innovation, Harvard Business School Press, Boston*.

Gloor, Peter A, Krauss, Jonas, Nann, Stefan, Fischbach, Kai, & Schoder, Detlef. (2009). *Web science 2.0: Identifying trends through semantic social network analysis.* Paper presented at the Computational Science and Engineering, 2009. CSE'09. International Conference on.

Granovetter, Mark S. (1973). The strength of weak ties. *American journal of sociology*, 1360-1380.

Grewal, Rajdeep, Lilien, Gary L, & Mallapragada, Girish. (2006). Location, location, location: How network embeddedness affects project success in open source systems. *Management Science, 52*(7), 1043-1056.

Henneberg, Stephan C, Gruber, Thorsten, & Naudé, Peter. (2012). Services networks:

Concept and research agenda. *Industrial Marketing Management, 42*(1), 3-8.

Hilkert, Daniel, Benlian, Alexander, & Hess, Thomas. (2011). *The Openness of Smartphone Software Platforms–A Framework and Preliminary Empirical Findings from the Developers' Perspective.* Paper presented at the Annual Conference of the Gesellschaft für Informatik (INFORMATIK 2011). Springer (Lecture Notes in Informatics), Berlin, Germany.

Hwang, Junseok, Altmann, Jörn, & Kim, Kibae. (2009). The structural evolution of the Web 2.0 service network. *Online Information Review, 33*(6), 1040-1057.

Kibae, Kim. (2011). *An Analysis of the Openness of the Web2.0 Service Network Based on Two Sets of Indices for Measuring the Impact of Service Ownership*.

Kim, K., Lee, W.-R., & Altmann, J. (2013). *The trace of position of SaaS entering the SaaS network*. Paper presented at the In Proceeding of the 21st ECIS, Utrecht, The Netherlands.

Krackhardt, David, & Stern, Robert N. (1988). Informal networks and organizational crises: An experimental simulation. *Social psychology quarterly*, 123-140.

Lévy, Pierre. (2010). From social computing to reflexive collective intelligence: The

IEML research program. *Information Sciences, 180*(1), 71-94.

Leimeister, Stefanie, Böhm, Markus, Riedl, Christoph, & Krcmar, Helmut. (2010). The business perspective of cloud computing: Actors, roles and value networks. *ECIS 2010 Proceedings, 56*.

Maglio, Paul P, Srinivasan, Savitha, Kreulen, Jeffrey T, & Spohrer, Jim. (2006). Service systems, service scientists, SSME, and innovation. *Communications of the ACM, 49*(7), 81-85.

Milgram, Stanley. (1967). The small world problem. *Psychology today, 2*(1), 60-67.

Newman, Mark EJ. (2001). Clustering and preferential attachment in growing networks. *Physical Review E, 64*(2), 025102.

Newman, Mark EJ. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary physics, 46*(5), 323-351.

O'reilly, Tim. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications & strategies*(1), 17.

Qu, Renjun, & Ye, Qiongwei. (2010). Service Ecosystems of cloud computing. *Atlantis Press*, 61-70.

Rimal, Bhaskar Prasad, Choi, Eunmi, & Lumb, Ian. (2009). *A taxonomy and survey of cloud computing systems.* Paper presented at the INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on.

Smith, Roger. (2009). Computing in the cloud. *Research Technology Management, 52*(5), 65.

Tsai, Wenpin. (2001). Knowledge transfer in intraorganizational networks: Effects of network position and absorptive capacity on business unit innovation and performance. *Academy of management journal, 44*(5), 996-1004.

Valverde, Sergi, & Solé, Ricard V. (2007). Self-organization versus hierarchy in open-source social networks. *Physical Review E, 76*(4), 046118.

Wagner, Caroline S, & Leydesdorff, Loet. (2005). Network structure, self-organization, and the growth of international collaboration in science. *Research policy, 34*(10), 1608-1618.

Watts, Duncan J, & Strogatz, Steven H. (1998). Collective dynamics of ‘small-world’networks. *nature, 393*(6684), 440-442.

Weiss, Aaron. (2005). The power of collective intelligence. *networker, 9*(3), 16-23.

Youseff, Lamia, Butrico, Maria, & Da Silva, Dilma. (2008). *Toward a unified ontology of cloud computing.* Paper presented at the Grid Computing Environments Workshop, 2008. GCE'08.

Appendix 1: Sample of Application List

App Name	Provider	Released Date
AccountNewsFeed	SalesforceLabs	1/30/2008
ActionPlans-v3-Unmanaged-EE,UEandDE	SalesforceLabs	6/4/2010
Activities(Lite)forChatter	RadialWeb	6/27/2011
Adobe®EchoSign:GlobalElectronicSignatureService	EchoSignfromAdobeSystemsIncorporated	10/6/2006
AffiliationsfortheNonprofitStarterPack	Salesforce.comFoundation	10/15/2010
AJAXTools	SalesforceLabs	1/31/2007
ApexTutorialsWinter'12	SalesforceLabs	12/1/2011
AppExchangeDashboardPack	SalesforceLabs	9/4/2007
AppirioCloudSyncforGoogleApps	Appirio	11/29/2011
AttachmentManager	SalesforceLabs	3/19/2007
AutoCompleteComponentforSalesforce	SolcomInternacionalS.A.deC.V.	6/27/2011
BarcodeZone	ApexCloud	2/14/2011
BirthdayReminder	InteractiveTies	3/11/2009
CampaignMembershipReport	SalesforceLabs	1/8/2006
CampaignTimelineCalendar	SalesforceLabs	2/26/2007

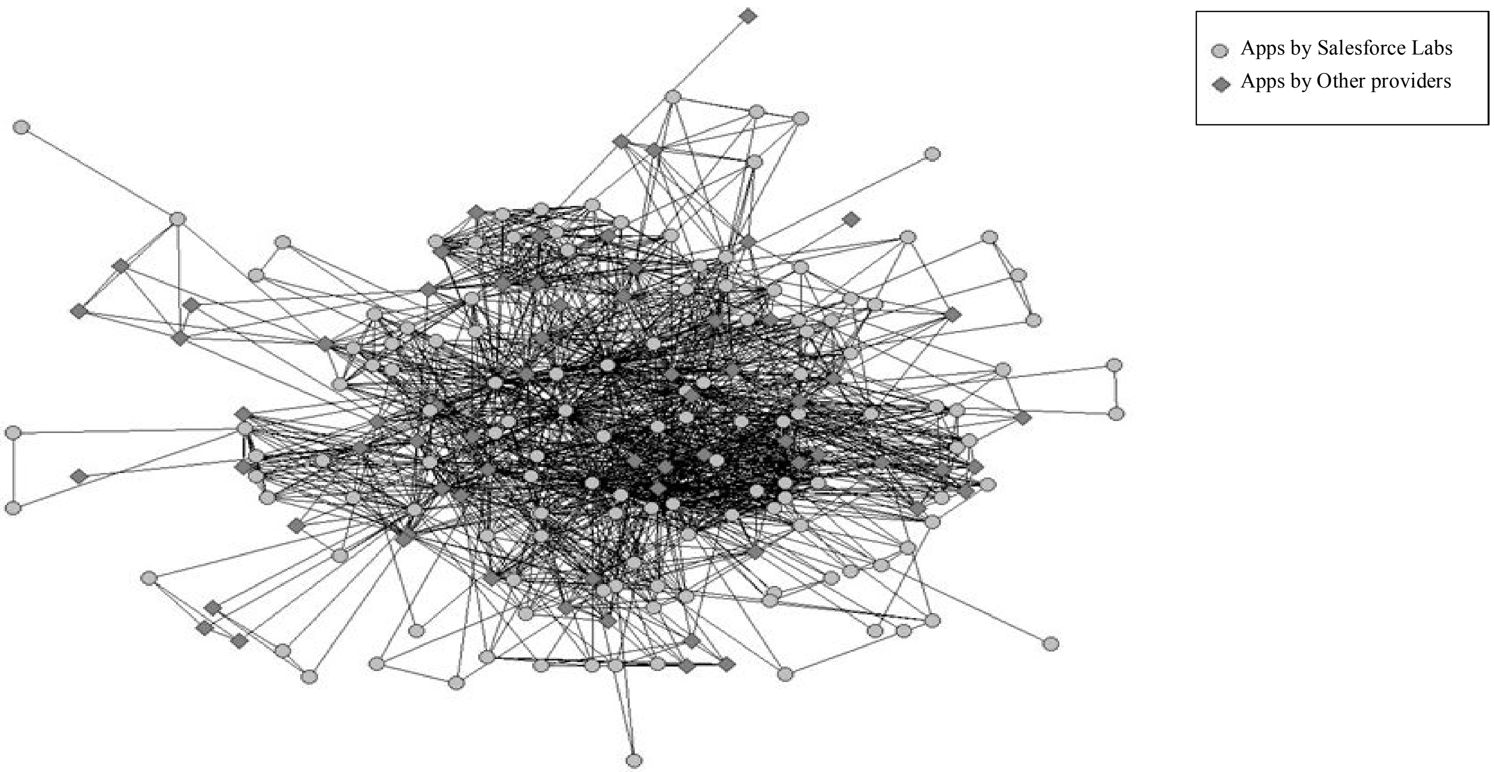
Appendix 2: Sample of Developer List

User	Installed Applications			
Garry Polmateer	Action Plans	Conga Composer		
Haribabu Amudalapalli	Activities (Lite) for Chatter	Find Nearby		
Digvijay Singh	Apex Tutorials Winter '12	Chatter Combo Pack - Managed		
Mustafa Turab Ali	Appirio Cloud Sync for Google Apps	CMSForce 2	Geopointe: Mapping&Geo-Analytics	
Bruno Lube	Barcode Zone	TuBarcode by Scout		
Dave Affentranger	Campaign Call Down Manager	Simple Quote v1.0		
JoAnn Culbertson	Create Opportunity & Quote	Visual Workflow Getting Started Pack	Graphics Pack	Marketing Calendar

Appendix 3: Sample of Link list

Source App	Target App	Num of Apps	Developer
Action Plans	Conga Composer	2	Garry Polmateer
Activities (Lite) for Chatter	Find Nearby	2	Haribabu Amudalapalli
Apex Tutorials Winter '12	Chatter Combo Pack - Managed	2	Digvijay Singh
Appirio Cloud Sync for Google Apps	CMSForce 2	3	Mustafa Turab Ali
Appirio Cloud Sync for Google Apps	Geopointe: Mapping & Geo-Analytics	3	Mustafa Turab Ali
CMSForce 2	Geopointe: Mapping & Geo-Analytics	3	Mustafa Turab Ali
Barcode Zone	TuBarcode by Scout	2	Bruno Lube
Campaign Call Down Manager	Simple Quote v1.0	2	Dave Affentranger
Create Opportunity & Quote	Visual Workflow	4	JoAnn Culbertson
Create Opportunity & Quote	Graphics Pack	4	JoAnn Culbertson
Create Opportunity & Quote	Marketing Calendar Free	4	JoAnn Culbertson

Appendix 4: Applications Network



Appendix 5: Sample of Network Position Analysis Results

AppName	Provider	DegCtr	BtwCtr
FieldTrip	Qandor	87	2317.309
CongaComposer	Conga	86	2976.175
MassUpdateAndMassEditFromListView	SalesforceLabs	79	248.5531
AppirioCloudSyncforGoogleApps	Appirio	76	2312.737
InsideViewFREE	InsideView	60	3094.161
CloudConverterforForce.com	ModelMetrics	47	568.4163
MassDelete	SalesforceLabs	45	600
EtheriosEasyPage-PageGenerator	Etherios	43	394.804
GroupMaster	SalesforceLabs	42	395.2192
FindNearby-Accounts,Contacts,Leads-Managed,PE/EE/UE/DE	SalesforceLabs	41	502.4574
LinkedInSalesNavigatorforSalesforce	LinkedIn	38	487.2074
FliptopSocialProfiles:LinkedIn,TwitterandFacebookProfiles	Fliptop	37	370.8345
MassUpdateContactAddress	X-SquaredOnDemand	36	346.7169
LeadScoring	SalesforceLabs	36	569.8625
TimbaSurveysbyAltimetrik	Altimetrik	35	286.8271

Abstract (Korean)

IT기술의 발전으로 소비자들이 직접 혁신에 참여하도록 촉진하는 새로운 비즈니스 모델이 떠오르고 있다. 이는 인터넷을 통해 맞춤형 소프트웨어를 공급하는 Software-as-a-service (SaaS)의 출현과 함께 발전하고 있다. 이전의 연구들은 네트워크의 구조와 진화 패턴, 네트워크 특성과 혁신 성과와의 관계를 연구하였으나, 내부적인 방법에만 초점을 두어, 제 3자 개발자들과 사용자들을 끌어들이려는 플랫폼 제공자의 노력은 고려되지 않았다. 따라서 본 연구에서는 네트워크 측면에서의 서비스 생태계 내의 집단 혁신 안에서, 플랫폼 제공자가 제 3자 개발자들과 사용자들의 참여를 촉진하기 위해 들인 노력을 연구한다. 배경 개념을 소개한 후, 서비스 네트워크의 성장, 허브와 플랫폼 제공자의 개방성을 세일즈포스닷컴의 AppExchange와 Developer Market에서 얻은 실증적 데이터로 분석하였다. 분석 결과에서 플랫폼 제공자가 초기 서비스 네트워크의 성장에 기여하였으며, 후에는 제 3자 제공자들이 더 많이 참여하고 있는 것을 보여준다. 본 연구는 학술적 측면에서 집단 지성을 통해 혁신에서 서비스 네트워크의 진화를 촉진하는 플랫폼 제공자의 역할을 나타내고, 산업적 측면에서 플랫폼 리더들에게 그들의 서비스 생태계를 진화시키기 위해서는 제 3자 개발자들과 사용자들을 참여시키기 위한 노력을 해야 됨을 시사한다.

주요어 : Software-as-a-Service, 세일즈포스닷컴, 서비스 연결망, 오픈 이노베이션, 플랫폼 리 더, 집단 지성
학 번 : 2011-23443