Platform Provider Roles in Innovation in Software Service Ecosystems

Kibae Kim and Jörn Altmann

*Abstract***—Following a new innovation strategy, software vendors move their software onto their software service platforms and open up their platforms to third-party software service vendors. Although many studies state that enlarging the scope of software service offerings is the goal of the platform providers, only a few studies have focused on the roles that the platform providers take on to achieve the goal. These studies identified that the platform providers not only manage the platform but also promote and regulate third-party software. In this article, we extend this research by analyzing the activities of how platform providers promote innovation. For the analysis, we use empirical data about software services gathered from AppExchange of Salesforce. The analysis identifies the clusters and positions of software services of Salesforce and third-party vendors in the software service network. Our analysis results show that Salesforce promotes innovation by provisioning software services that are in core positions or that are bridges between service clusters. Third-party vendors release software services that are complementary to those of Salesforce. Overall, the results suggest that platform providers need to position strategically their software services to build successful software ecosystems, and that research on innovation needs to analyze the roles and efforts of the platform providers in detail.**

*Index Terms***—Open innovation, platform leadership, Salesforce.com, social network analysis, software industry, software-as-a-service, software services.**

I. Introduction

ONE of the most interesting software innovation strategies is that software vendors create software service platforms for offering their software as a service and open up their platforms to attract third-party software vendors. Previously, software vendors sold their software with on-premises, perpetual software license. By moving toward software service platforms, software vendors benefit from the collective intelligence of platform users [1], [2]. Collective intelligence enables the creation of new ideas through the voluntary sharing of knowledge and innovation resources of the platform users with each other, stimulating innovation beyond the individual stakeholder capabilities [1], [3]–[5]. This way, software can be reused to create new software, which would not emerge due to the limitation in resources otherwise. For example, if it is possible to reuse software, a single software vendor, who has to focus on implementing a few features due to the development cost for software [6], can focus on the implementation of new innovative features instead of reimplementing existing features.

The key to successful platform strategies is to attract third-party vendors developing applications and end-users (consumers) consuming these applications. The revenue generated from their transactions can be shared between the different stakeholders of the platform ecosystem [7], [8]. To design such a platform architecture and the corresponding business model, a platform provider needs to address the technological and business challenges. From a technical viewpoint, a platform provider needs to define the software services and the interfaces between them. In particular, a platform provider needs to distinguish the core software services, which it wants to offer, from those software services that it wants to leave to third-party vendors [9]–[11]. From a business viewpoint, the platform provider needs to identify the commercial side and the subsidy side of the market, regulate their transactions [12], [13], and to control the openness of the innovation resources on the platform [14], [15].

Prior research assumes that third-party vendors vigorously create new software applications if a platform provider designs a well-defined architecture and business model. However, recent studies on platform-related innovation suggest that innovation on a platform requires an assiduous involvement in the activity of third-party vendors [10], [16]. Especially, a platform provider needs to release applications of its own to promote third-party vendors' innovations. For example, Apple has released software applications (i.e., map and payment applications [16]) that are fundamental to the applications that third-party vendors develop. Gawer and Cusumano [10] called this platform provider's designing of architecture and business model "coring" and its continued endeavor of promoting third-party vendors' innovations "tipping." Furthermore, while a lot of previous studies introduced theoretical propositions and empirical findings on platform architectures and business models of platforms, only few studies investigated the roles of platform providers related to regulation and promotion. Among those, the activities of a

Manuscript received November 24, 2018; revised February 13, 2019 and August 25, 2019; accepted October 1, 2019. Date of publication January 22, 2020; date of current version June 8, 2022. This work was supported in part by the Global Research Collaboration Program with the World Economic Forum for the Fourth Industrial Revolution (A0801030001) of the Ministry of Science and ICT, in part by the National Research Foundation of Korea under Grant NRF-2019R1F1A1058487 and Grant NRF-2016R1A6A3A11932416 that is funded by the Korea government (MSIT), and in part by the Institute of Engineering Research at Seoul National University, which provided research facilities. Review of this manuscript was arranged by Department Editor D. Meissner. *(Corresponding author: Jörn Altmann.)*

K. Kim is with the Korea Policy Center for the Fourth Industrial Revolution, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: kibaejjang@gmail.com).

J. Altmann is with the Technology Management, Economics, and Policy Program, College of Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: jorn.altmann@acm.org).

Digital Object Identifier 10.1109/TEM.2019.2949023

platform provider on stimulating participation of its third-party vendors (i.e., as a source of collective intelligence) in innovation has not fully been investigated.

The objective of this article is to address this shortcoming by determining the activities of platform providers. In particular, the objective is to identify how the platform providers participate in the innovation process within a software service ecosystem by investigating the position of software services released by platform providers and third-party vendors within an entire software service network, within the clusters of services of a software service network, and between the clusters.

To achieve this objective, we investigate the innovation on a software service platform with respect to the structure of its software service network. As a network structure emerges through the resources available, the value of resources, and the activities of stakeholders, the analysis of a network structure (i.e., the positions of resources within the network) can reveal important stakeholders or the important characteristics of resources. For our purpose, the network structure analysis is intended to reveal the additional mechanism behind innovations on software service platforms. The software service network analyzed in this research is defined as a tuple of a set of nodes and a set of links. While each node represents a software service, a link between a pair of nodes denotes the existence of a complementary relationship between the software services corresponding to the nodes. Social network analysis is conducted using empirical data gathered from AppExchange, an open platform for customer relationship management (CRM) software, which is managed by Salesforce.com[1] [17]. In particular, we measure the degree centrality of software services within the entire software service network, the z-score of software services within clusters of the software service network, as well as the connectivity of software services in a cluster with nodes outside the cluster [18]. Finally, we compare the network positions of services released by the platform provider with those released by third-party vendors, using the Wilcoxon Rank Sum test.

Our main findings show that the network positions of software services released by the platform provider are different from those released by the third-party vendors. That is, the software services of the platform provider locate at the center of the entire network compared to those of the third-party vendors. In addition to this, the platform provider releases software services locating at the core positions of service clusters and at the bridge positions that connect service clusters. Software services released by the third-party vendors are connected with software services released by platform provider, indicating that software services of the third-party vendors are likely to be complementary to software services seeded by the platform provider. In summary, the results suggest that a platform provider develops software services, which lead the innovation in the entire service ecosystem through seeding service clusters and demonstrating the value of combining services of different clusters.

Our findings suggest both academic and managerial implications. From an academic perspective, our findings suggest that the role of a platform provider should be refined. Prior art proposed that opening the resources of a platform provider to third-party vendors could lead to their participation in innovation on the platform. Our findings imply, instead, that providing software services for seeding a cluster of services could be added to the roles of a platform provider. This extension of the platform provider role leads to a managerial implication. While prior research suggests that a platform provider implements strategies for developing an ecosystem on the ground of its platform (e.g., separating the sides of a market [13] and tipping after coring [10]), our findings suggest to add the strategy of participating in the innovation that has been dedicated in prior research to the role of third-party vendors. Moreover, our findings help a platform provider to develop its ecosystem through releasing core software services and bridge software services that aggregate the third-party vendor software services (i.e., building a cluster of services) and guide the third-party vendors to connect those software services released by the platform provider.

The remainder of this article is structured as follows. Section II describes the conceptual background of the social network analysis that is used as the research method. The collection of the empirical data and the process of analysis are described in Section III. Sections IV and V lay out the analysis results and a detailed discussion of the analysis results, respectively. Section VI concludes this article.

II. Theoretical Background

A. Platform-Related Innovation

A new trend of innovation that has been identified in the early 2000s is that software leaders form “platform-based software ecosystems” [19]. A platform provides a base of software components (i.e., software applications) that support core functions and can be extended through external software components that use the standardized interfaces of the platform [19], [20]. These software component extensions can even be modified by developers for creating further extensions, avoiding complexity due to the interoperability of these services [21]. If extensions are added to a platform, the extensions not only complement the overall offerings for the platform users but also compete with other extensions, forming a software ecosystem [7], [19], [22], [23].

If a platform provider opens access to its innovation resources, platform customers (i.e., platform end-users and third-party vendors) can utilize them for their innovations [7]. The benefits are twofold. On the one hand, the platform provider harnesses the collective intelligence of third parties, achieving its innovation at a low cost [1]. Thereby, the platform provider gains benefit through the extension of the scope of their software applications [8]. On the other hand, the third-party vendors might collaborate with each other without any benefit to a platform provider [26]. To address this, some platform providers allow access to basic functions of their platform at no cost, while functions enabling sophisticated collaborations are only available under a commercial license (see “Security Check Fee” in Fig. 1). This allows them to gain revenue without limiting the innovation

[1] [Online]. Available: http://www.salesforce.com

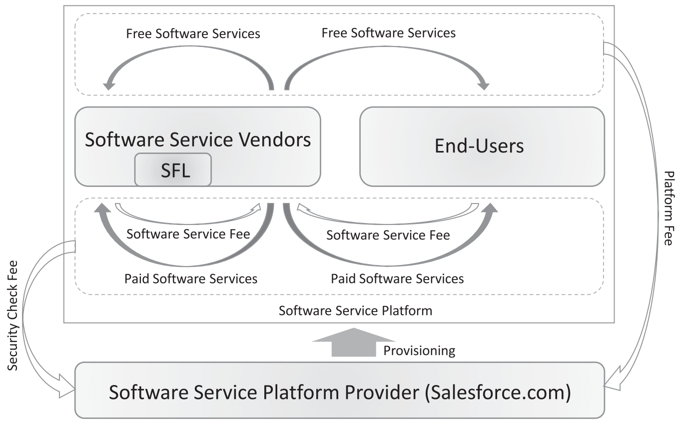

Fig. 1. Software service ecosystem that is based on Salesforce.com.

through collective intelligence [12]. The success of a platform-based software ecosystem relies on how vigorous the third-party vendors participate in the innovation [24], [25].

Prior research emphasized that the roles of platform providers are not only about making their resources available but also designing an architecture and a business model for its platform [8], [9], [11], [14], [15], [27], [66], [67], [68]. The roles of the platform providers can be distinguished into the technology-related and business-related roles. With respect to technology-related roles, a platform provider designs the architecture of its platform through the definition of components (i.e., "components of a system in which elements are densely connected") and the interfaces between the components [9], [11]. With respect to business-related roles, a platform provider determines means to attract end-users and motivate third-party vendors to innovate in a "multisided market" [14]. One of the key instruments is to adjust the openness of the platform by balancing the platform provider's power for charging and the third-party vendors' environment for innovating [15].

Additionally, platform providers participate in developing new software applications on their platforms. For this, Gawer and Cusumano [10] proposed two strategies for platform leadership (i.e., "coring" and "tipping"), addressing the technological and business challenges to promote third-party vendor participations. That is, the development of software applications are not only performed by the third-party vendors but also by the platform providers. Furthermore, Ghazawneh and Henfridsson [16] showed in their case study of Apple iPhone that a platform provider releases applications that can be seeds for developing third-party vendors' applications so that the diversity of applications enhances. In our case, Salesforce.com, which owns the software services platform, also participates as a software vendor in the innovation (see Fig. 1).

In summary, a platform can provide an environment for "open innovation," so that the knowledge permeates across the organizational boundaries [3], [28]. Third-party vendors access the resources of a platform provider for their innovation, while the outflow of their resources permits spin-off innovation on the provider platform. The incentives for the platform provider is the additional revenue from selling additional software services through the creation of a variety of software services and a critical mass of platform users. The critical mass, in turn, provides incentives to the third-party vendors to participate in the open innovation platform. Without this critical mass, third-party vendors cannot generate sufficient revenue to cover their software development cost [13]. Furthermore, subsidizing one side of the market (in order to achieve a critical mass) does not mean that a platform provider incurs a loss. The platform provider can gain sufficient revenue from the other side of the market (i.e., the money side of the market) through economic activities between the two sides [13], [15], [29], [30].

B. Position of Nodes in a Network

Prior research pointed out that innovation could depend on the system rather than individuals involved [31], [32]. This idea can also be applied to open innovation on software service platforms, which cultivate innovations by providing the possibility of combining innovation resources (i.e., software services) through third-party vendors. Following this idea, previous studies started to examine software services from a network perspective [33], [34], [69]. Defining a node and a link to represent a software service and a new software service codevelopment, respectively, the authors described the topology of a software service network and traced the change of positions of the nodes in the network. Different to this approach, we represent a software service as a node and the coinstallation of a pair of software services within the account of a platform user as a link between the two software services [71]. Our definition follows two assumptions: First, a platform user utilizes a set of software services, which complement each other in addressing the need of the platform user. Second, a cooccurrence of software services simply means that a technological proximity or a cognitive proximity between software services exists.

As a network structure emerges depending on the resources available, the value of resources, and the stakeholder activities, the analysis of a network structure (i.e., the analysis of the positions of resources within the network) can reveal stakeholders or the characteristics of resources, which are important to the innovations within the system. Therefore, network structure analysis (using social network analysis) could reveal mechanisms behind innovations on software service platforms, which cannot be detected otherwise.

As a platform provider can access its platform earlier than any third-party vendor can, it has a first-mover advantage in developing software services. Therefore, a platform provider can release a set of software services, which represent core software services that provide value to the platform users. As third-party vendors do not have this advantage, software services of third-party vendors rationally build their software services with those of the platform provider. As a result, software services released by the platform provider might be used by customers more often than those developed by the third-party vendors. Consequently, it is expected that the software services developed by the platform providers are probably located more central than those of the third-party vendors. Based on this, we establish the first hypothesis:

Hypothesis 1. The software services released by a platform provider locate at more central positions in a software service network than those released by the third-party vendors.

C. Network Position Within a Cluster

Network studies have shown that some nodes in a network are connected more densely than other nodes. These groups of densely connected nodes are called clusters [35]–[37]. The process of forming clusters is affected by the preferences of people for similarity [38]. Examples of clusters can be found in networks about the physical distance between houses [39], academic background of researchers [36], and demographic properties (e.g., gender and race) of people [40], [41].

Innovation studies focus on the process through which knowledge is created within and between communities (i.e., clusters) [42]–[44]. Knowledge is constructed within a community, in which agents share their academic backgrounds, norms, perspectives, and methodologies [45], as well as between communities. For example, innovation is achieved if a community with a problem hard to solve finds a solution to this problem with the knowledge from another community [43]. In this process, an agent with a connection to the other community obtains the knowledge from the other community and disseminate it to its own cluster [42], [46]. This agent locates at a position that mediates the communities (clusters) in a network. This position in the network structure is called "structural hole" [42], [47].

Similar to those social networks, software services tend to be connected with other software services that share identical attributes [48]. Software services within the same service category are more likely to be combined than software services belonging to different service categories. Furthermore, software services are likely to be connected with those software services that have been released by the same vendor. Following Kim *et al.* [48], we also assume that a software service network contains several clusters, in which software services are closely connected. Innovation is achieved within these clusters of the software service network in the same way as in traditional social networks. If a platform provider intends to promote innovation of third-party vendors on its platform, one of the best strategies is to develop software services that build their own communities. In this case, the software services released by the platform provider would be at the center of clusters. Following this line of argument, we establish the following hypothesis:

Hypothesis 2. The software services released by the platform provider locate at more central positions within a cluster than those released by the third-party vendors.

D. Network Position Across Clusters

A network evolves as clusters emerge, merge, or disappear. That is, a network adapts to a changing environment [44]. The evolution of a network depends on the interactions between the clusters. An empirical study shows that the probability of an organization's survival increases if subgroups (i.e., clusters) in the organization interact with each other [49]. This correlates with innovation processes. Innovation is mainly achieved through interactions within clusters [49], while solutions to hard problems of a cluster might be found in another cluster [43], [46].

The prior research suggests that, as it is impossible for a software service platform provider to catch each innovation trend in detail and to invest in all potential services, a platform provider needs to attract third-party vendors that take over this task [3], [7]. However, as third-party vendors would face cognitive limitations in reusing and recombining unfamiliar software services [50], they might be captivated in providing slightly advanced software services within a service clusters initiated by the platform provider. That is, it is the role of the platform provider to open the opportunity of bridging distant clusters of software services. In this environment, third-party vendors can develop software services complementary to the bridge services released by the platform provider [7], [24]. Based on this discussion, we suggest the following hypothesis:

Hypothesis 3. The software services released by the platform provider are more likely to be connected with the nodes outside their clusters than those software services released by the third-party vendors.

III. Methodology

A. Salesforce.com's Offering of a Software Service Platform

Salesforce.com focuses on CRM systems. It provides its CRM software as software services (also known as Software-as-a-Service (SaaS)) so that end-users can outsource their CRM for their businesses. Furthermore, it provides a platform with its CRM services for third-party vendors to customize their services and develop new services on it [17], [51]. In other words, third-party vendors participate in the innovation and utilize the innovation output from the software service platform provided by Salesforce.com. It is similar to Apple iOS and Google Android, which are platforms for developing and utilizing mobile apps. Therefore, Salesforce.com, its third-party service vendors, and their end-users form a software service ecosystem (or "software ecosystem" as defined in [52] or "digital ecosystem" as defined in [26]). In this software service ecosystem, innovation is performed by the cooperation and interaction among the stakeholders.

Fig. 1 illustrates the software service ecosystem of Salesforce.com, which is based on the software service platform, AppExchange, of Salesforce.com. The ecosystem comprises a software service platform provider (Salesforce.com), third-party software service vendors, the Salesforce Lab (SFL) software service vendor, and end-users, who consume software services created by software service vendors. Platform users (i.e., software service vendors and end-users) are charged based on the usage of the software service platform. Software service vendors can offer their software services to the platform users free-of-charge or for a fee. That means, a software service vendor can also utilize the software services created by other vendors. If a vendor charges a fee to the platform users, Salesforce.com requests a security check fee from the software service vendor. Salesforce.com also creates software services, using the name SFL. All software services of the platform are enlisted in a directory. If customers install software services under their accounts to support their

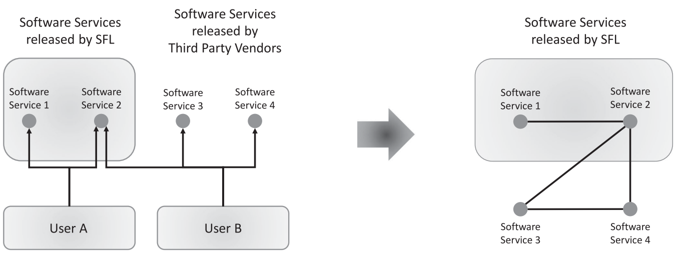

Fig. 2. Example of a software service network constructed from the data of AppExchange of Salesforce.com.

own business, they are charged for the service usage according to the business model of the software service vendor.

B. Data Collection

The data for the empirical analysis have been collected from the service installation and service developer information pages of AppExchange of Salesforce.com.[2] Each entry on the information page about a software service contains information about the provider of the service, the software service name, the release date, the pricing plan, the service categories, the reviews, and the rating of the software service. The information that is used for the analysis includes the software service name, the provider of the service, and the release date. The information about the software service coinstallations comes from the information that has been made available by the software service vendors. The information about software service coinstallations comprises the name of the vendor (i.e., the Salesforce.com account) and the list of services that the software service vendor has installed.

Based on this data, the software service network has been constructed. The software service network is defined as a tuple of a set of nodes and a set of links. While the nodes represent the software services operating on the Salesforce.com platform, the links represent the software service coinstallations. That means, the link between a pair of nodes represents the complementary relationship between two software services as used by a platform user (i.e., service vendor or end-user). That is, all software services used by a platform user are complementary to each other. Fig. 2 illustrates an example of this kind of the software service network. The example shows SFL, which is operated by Salesforce.com (left-hand side of Fig. 2) and has released “mass update and mass edit from list view” and “survey force” (corresponding to Software Service 1 and Software Service 2, respectively). Two third-party software service vendors released “inside view free” and “conga composer” (corresponding to Software Service 3 and Software Service 4, respectively). Platform user A installed Software Service 1 and Software Service 2, and platform user B installed Software Service 2, Software Service 3, and Software Service 4. Then, the example leads to a software service network with 4 nodes and 4 links (i.e., links between Software Service 1 and Software Service 2, Software Service 2 and Software Service 3, Software Service 3 and Software Service 4, and Software Service 4 and Software Service 2 (right-hand side of Fig. 2)).

[2][Online]. Available: http://appexchange.salesforce.com

C. Identifying Clusters in a Network

A cluster is a subset of a network, in which the nodes are more similar than the nodes outside the cluster [36], [37], [53], [54]. We use the simulated annealing algorithm to detect the clusters in the software service network [54]. This algorithm stochastically finds globally optimal clusters using metaheuristics so that the clusters have a high modularity with respect to the number of links that nodes generate within and across the clusters [54]. This approach yields clusters whose nodes are densely connected within the clusters and sparsely connected across the clusters.

D. Measuring the Network Position

Centralities measure the network position of nodes. There are a variety of centralities including the degree centrality [55]. As the degree centrality of a node measures the number of nodes connected to the node, it is a most intuitive and simple way to measure how deep the node is embedded in its network. The degree centrality is normalized by the maximum number of links $(g-1)$ that it can possibly have in a network of size g. Assuming that $a_{ij}=1$ represents the link between two nodes i and j, then the normalized degree centrality d_i of node i in a network of size g is defined as

$$d_i = \sum_{j=1}^{g} \frac{a_{ij}}{g-1}$$

assuming that $a_{ij}=0$ if node i and node j are not connected. If a node i is isolated from any other node, the degree centrality is $d_i = 0$.

Based on this definition of degree centrality, we define the z-score z_i^C of a node i within a cluster C, as proposed by Guimerà and Amaral [54]. For this, we define the degree centrality d_i^C of node i within a cluster C to be the number of links to the other nodes within the cluster (i.e., $\sum_{j\in C} a_{ij}$). Furthermore, for the cluster C containing node i, we define $\bar{d}_C$ and σ_{d_C} to be the average of all d_j^C with $j \in C$ and the standard deviation of all d_j^C with $j \in C$, respectively. Using these definitions, the z-score of node i can be calculated as

$$z_i^C = \frac{d_i^C - \bar{d}_C}{\sigma_{d_C}}.$$

In addition to this, the external connectivity of node i belonging to cluster C measures the number of links to nodes outside the cluster. It is normalized by the maximal possible number of links $(g-c)$ for a cluster C with size c in a network of size g. In detail, the normalized external connectivity $d_i^{\neg C}$ of node i belonging to cluster C is defined as

$$d_i^{\neg C} = \sum_{j\in\neg C} \frac{a_{ij}}{g-c}.$$

IV. Analysis Results

A. Data Description

The data collected show that about 1000 software service vendors registered on the AppExchange platform and 74 of those vendors published their information about their software

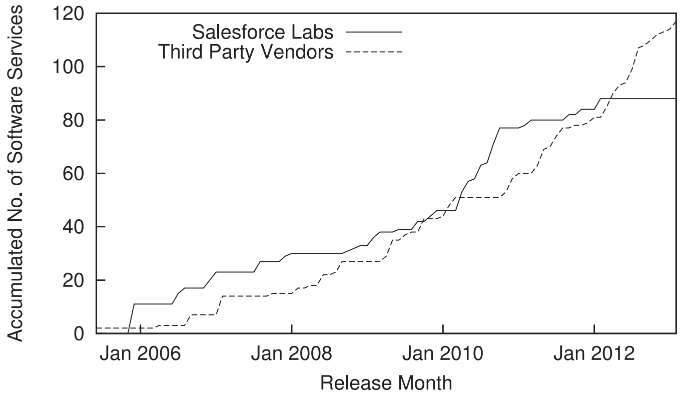

Fig. 3. Number of software services on Salesforce.com's AppExchange.

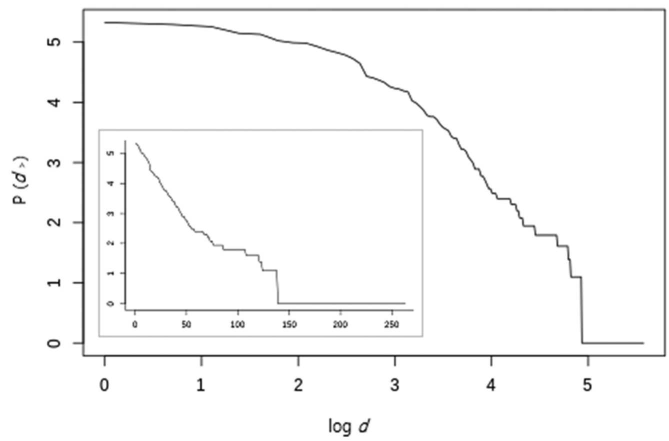

Fig. 4. Cumulative degree distribution in base-e log-log scales of the 206 software services (The graph in the small box depicts the same data in base-e log-linear scales.)

service installations. These are the users of software services released on the AppExchange of Salesforce.com. The 74 vendors utilized 206 software services. These 206 software services were created by 99 different vendors. Among the 206 software services, 88 software services were released by SFL, a brand of Salesforce.com. The remaining 118 software services were released by 98 third-party software service vendors. The total number of new software services per month increased steadily during the study period from July 2005 to April 2013 (see Fig. 3).

In detail, Fig. 3 depicts the trend of the number of software service released by SFL and the trend of the number of software service released by the third-party vendors. The number of software services, which have been released by SFLs, steadily increased from 11 software services in January 2006 to 77 software services in November 2010 but slowed down since then. At the end of the study period, only 88 software services have been released in total. In other words, the rate of contribution of SFL (i.e., the number of software services released compared to the total number of software services) decreased gradually over time.

The increase in the number of software services, which have been released by the third-party vendors, shows a different trend. The number of software services released by the third-party vendors was two software services in January 2006 and 51 software services in November 2010. Afterward, it increased faster than the number of software services released by SFLs. The third-party vendor contributions surpassed the number of software services released by SFLs in May 2012. The number of software services of the third-party vendors is 118 at the end of the study period.

B. Network Topology

For the software service network that has been formed, Fig. 4 shows the cumulative distribution of the degree centrality in base-e log–log scales (and log-e log-linear scales in the small box). The cumulative degree distribution seems to decay by a power function with a concavity. The topology of the Salesforce.con software service network looks different from a network with a usual power-law distribution, considering that the cumulative distribution of a power law fits to a linear line in log-log scales [56].

To check the similarity of the decay with an exponential function, we fitted the cumulative degree distribution in a log-linear scale (the graph within the small box of Fig. 4). The results show that the cumulative degree distribution decays following an exponential function in the low-degree area but not in the high-degree area. The high-degree area contains "mass update and mass edit from list view," "field trip," "appirio cloud sync for google apps," "mass delete," and "conga composer." They have no common property with respect to the service category and payment schemes. The categories of these services are different from each other. Two of them are charged for, and the remaining three are free of charge. However, it is interesting that these software services have been released by the third-party vendors.

The topology results suggest that the degree distribution of the service network can be characterized as a combination of a power-law function and an exponential function. The power-law distribution is generally formed through the evolution of the network. The evolution is based on a positive feedback to the degrees in previous time periods, i.e., through a preferential attachment [57]. An exponential distribution is the result of a random selection process [35]. Consequently, we can state that the software service ecosystem of Salesforce.com evolves due to the random selection of software services and a selection that is driven by a few hubs.

C. Clusters in the Software Service Network

Using the simulated annealing algorithm [54], we identified ten clusters that yield the modularity 0.40641. Fig. 5 depicts the descriptive statistics of those clusters in two dimensions: size of cluster and density within cluster. The size of a cluster represents the number of nodes belonging to the cluster. It varies from 2 to 37 for the 10 clusters identified. The density within a cluster, which is defined as the number of links within the cluster divided by the total number of links possible, varies between 0 and 10.0962 percent. The density within a cluster appears to

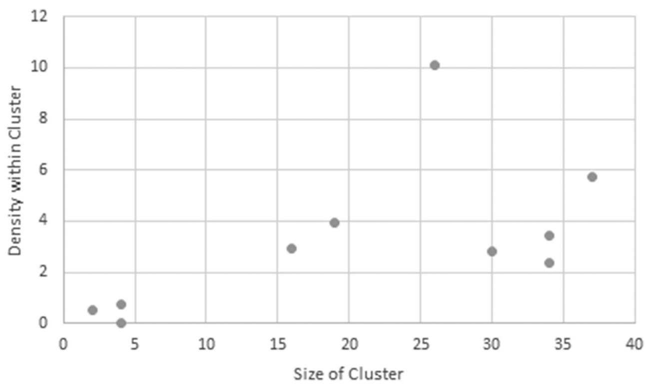

Fig. 5. Descriptive statistics of clusters.

TABLE I
RESULTS OF THE NORMALITY TEST (KOLMOGOROV–SMIRNOV STATISTICS) AND THE HOMOSCEDASTICITY TEST (LEVENE STATISTICS)

Variable	Group	Kolmogorov-Smirnov statistic [a]	Levene statistic [b]
Degree centrality	SFL	0.5000 ***	1.6509
	TPV	0.4950 ***	(1, 204)
z-score	SFL	0.1028	5.9833 **
	TPV	0.1845 ***	(1, 204)
External connectivity	SFL	0.5000 ***	2.3300
	TPV	0.4800 ***	(1, 204)

[a] Degree of freedom is 88 for SFL and 118 for TPV.
[b] Parentheses depict the degree of freedoms.
*Significant at 10% level.
**Significant at 5% level.
***Significant at 1% level.

TABLE II
WILCOXON RANK SUM TEST RESULTS FOR SFL AND TPV

Variable	Group	Mean	St. dev.	W statistic [a]
Degree centrality	SFL	0.1273	0.1772	3831 ***
	TPV	0.0900	0.1110	
	Total	0.1032	0.1389	
z-score	SFL	0.2763	1.2358	3955 ***
	TPV	-0.1242	0.7841	
	Total	0.0177	0.9844	
External connectivity	SFL	0.0897	0.1376	3968.5 ***
	TPV	0.0616	0.0895	
	Total	0.0715	0.1095	

*Significant at 10% level.
**Significant at 5% level.
***Significant at 1% level.

have a positive correlation with the size of a cluster except for an outlier whose size and density are 26 nodes and 10.0962 percent, respectively. One cluster shows 0 density. There are no links among the 4 nodes within the cluster, as the simulated annealing algorithm of the article presented in [54] aggregates the isolated nodes in one cluster.

D. Hypotheses Tests Results

To compare the network position of the software services released by the platform provider (group SFL) with those of the third-party vendors (group TPV), we split the data (206 software services in total) into two sets (i.e., the set consisting of a software service released by the platform provider and the set representing the software services released by the third-party vendors) and test their differences. Our dataset contains 88 software services in group SFL and 118 software services in group TPV.

Before comparing the two groups with respect to the degree centrality, z-score, and external connectivity, we test their normality and homoscedasticity to determine whether or not we can apply the algorithm of analysis of variance [58]. We implemented the Kolmogorov–Smirnov test for testing the normality of our empirical data [59] and Levene's test for the homoscedasticity [60]. The Kolmogorov–Smirnov test compares the cumulative distribution between the empirical dataset and the ideal normal distribution to determine whether the empirical dataset fits to a normal distribution as a null hypothesis. The Levene's test compares the variances of two samples to determine whether their variances are homogeneous as a null hypothesis.

Table I depicts the results of Kolmogorov–Smirnov tests and Levene's tests. The Kolmogorov–Smirnov statistics are significant for all indicators within a 1% significance level apart from the z-score indicator for group SFL. The results mean that, except for the z-score indicator for group SFL, the samples do not follow a normal distribution. The Levene statistic for comparing groups SFL and TPV for the z-score indicator is significant within a 5% significance level, while the Levene statistics for the degree centrality and external connectivity are insignificant. The results mean that the groups SFL and TPV have homoscedasticity in the degree centrality and external connectivity. In summary, it is hard to say that our datasets fulfill the normality and homoscedasticity conditions. Therefore, the parametric analysis of variance (or t-test for comparing two samples) is not recommended for our inference.

As an alternative method, we use the Wilcoxon Rank Sum test for testing the difference in the network position of software services between the platform provider (SFL) and the third-party vendors (TPV). The Wilcoxon Rank Sum test compares the sum of ranks in an ascending order between two groups to determine their rank sum is identical as a null hypothesis. If the two groups have different sample size, the algorithm builds pairs for according to the size of the smaller group. We apply this algorithm to the three indicators that we defined above (i.e., the normalized degree centrality in the entire network, the z-score, and the normalized external connectivity).

Table II shows the results of the Wilcoxon Rank Sum test. The Wilcoxon's statistics are considerably large for all indicators (i.e., for the degree centrality, the z-score, and the external connectivity), suggesting that the two groups (SFL and TPV) are significantly different within the 1% of significance level with respect to those indicators. In detail, the degree centrality of group SFL (average at 0.1273) is significantly higher than the one of group TPV (average at 0.0900). The results suggest accepting Hypothesis 1. Likewise, group SFL shows a larger z-score (average at 0.2763) and a larger external connectivity (average at 0.0897) than group TPV (average at -0.1242 in z-score and 0.0616 in external connectivity). Therefore, Hypothesis 2 and Hypothesis 3 are also accepted according to our analysis results.

V. Discussion

A. Theoretical Implications Derived From the Platform Providers Stimulating Interactions Between the Third-Party Vendors Through Their Vendor Role

From the traditional view of open innovation and management science, a platform provider could achieve sustainability if it develops an innovation ecosystem by opening up its innovation resources to third-parties and allowing third-parties to be a part of its innovation process [3], [7]. In this view, a platform provider allows its platform users to utilize its innovation resources, and the platform users develop goods complementary to the provider's goods. Moreover, the platform provider can increase the quality of its goods effectively by concentrating on the innovation of its core technologies and leaving the complementary goods developments to the platform users. Then, the quality and complementarity of the group of goods on the platform attract more end-users (customers) than it would without opening up resources [7]. Consequently, platform providers increase their profits.

While management science is interested in strategies of platform providers using an open innovation approach, network science has investigated mechanisms on how the platform users self-organize a software ecosystem through collective intelligence. It has been discussed that the evolution of the network follows an internal mechanism (e.g., preferential attachment mechanism and random rewiring mechanism) [35], [57]. Because of these internal mechanisms, the ecosystem of open innovation grows on a platform. The platform provider just supports the innovation of third-party vendors but does not contribute through the release of own software services. Networks, which are the result from these mechanisms, show specific topologies (e.g., scale-free topologies or random topologies). Examples of these networks are coauthorship networks [61], WWW [32], and the Internet [62]. Moreover, real networks contain clusters, in which the nodes are densely connected [44], [59], [63]. It means that the agents belonging to the same cluster normally interact more frequently than the agents belonging to different clusters. However, a significant innovation can only emerge if an agent connects to another agent in another cluster enabling the flow of new ideas between the different communities [46].

These previous studies do not directly connect research about the platform provider's side and research about the platform users' side. No research considered that a stakeholder could take roles on both sides. In management science, the platform users are only objects that move according to the strategy of the platform provider. In network science, the rules for forming a platform are already given, leaving no room for the provider to interfere with the interaction of the platform users. In real life, however, the platform providers continuously adapt their strategies to make their platforms more attractive. They open up their platforms and their development environments for the different platform users (i.e., third-party vendors, integrators, and end-users) [64]. Furthermore, they try to horizontally and vertically integrate their platforms with other platforms to compete with its rivals by achieving a critical mass [65] and enhancing their interoperability [65], [68].

Our analysis addresses this shortcoming by showing that a platform provider can also contribute to the network evolution and the initial setting of the network. The contributions of the platform provider motivate third-party vendors to participate in the innovation through the platform. Our analysis results show that a platform provider can lead the software ecosystem by providing software services, which locate at the core of clusters and/or bridge those clusters (e.g., the SFL software services "mass update" and "mass edit from list view"). These software services allow third-party vendors to bring in innovation through the release of software services that are complementary to those core and bridge software services. These results suggest extending the existing theories on management science and network science.

B. Managerial Implications for Software Service Providers

The results discussed are also relevant for providers of software service platforms from a managerial perspective. In most of the current models of software innovation through collective intelligence, the role of a platform provider is separated from the roles of third-party vendors. According to these models, third-party vendors share their innovation resources, reuse, and recombine them for innovation, while the platform provider simply prepares the environment for the interaction between the platform users. That means the ecosystem according to the existing open innovation theory allows third-party vendors to access the core technologies of platform providers [7]. The platform provider, who simply prepares the platform, does not lead the innovation through third-party vendors. It is assumed that they do not take on an active role in the innovation process.

The key to success of a software service platform is to attract platform users (i.e., software service vendors and end-users) [12]. End-users gather on a platform, if the quality of software services and the variety of complementary services on the platform are good. Software service vendors will come to the platform, if there is a large number of potential customers for their software services, guaranteeing the revenue that covers development costs [13], [66], [70]. It is based on the indirect network effect (i.e., the profit of the software vendors and the utility of the end-users) [61], [67]. A software platform will be sustainable, if a critical mass of platform users participates. Otherwise, it will fail. In particular, it is hard to attract software service vendors during the initial period due to the lack of platform end-users, and it is hard to attract end-users due to the lack of software services. In these cases, the platform does not reduce the innovation cost as open innovation promises [3]. However, as this is critical to the success of a platform, it is recommended that a platform provider actively participates in the innovation process initially. A platform provider should provision an initial set of its own software services, in an approach similar to those presented in the articles [10] and [16]. This way, they attract a critical mass of platform end-users and, in turn, a critical mass of software services.

Our empirical analysis results also indicate that a software service platform provider, who participates in the innovation process during the early time period of the platform, can provide incentives to platform end-users to join and, in turn, through the existence of platform end-users, make the platform attractive to third-party vendors. The platform provider Salesforce.com released more software services than the third-party vendors only in the beginning (see Fig. 3).

Furthermore, combining software services of a platform provider with software services of third-party vendors is a direct way of delivering software services of a platform provider to end users. Software services released by a platform provider became not only central in the entire network but also core in service clusters (i.e., service segments) and bridges between the clusters (see Table II). Those software services released by the platform provider guide the innovation of third-party vendors. Based on these results, the suggested strategy for platform providers is not only to take on the role of a vendor but also to guide third-party vendors in releasing software services around one or more segments of software services.

In conclusion, the strategy of a software service platform provider needs to be more comprehensive than the previous innovation studies in management science and network science expected. A platform provider does not just need to provide the platform, on which third-party vendors can offer and sell their services, and lead the innovation trend in the software ecosystem by forming the network but also needs to position strategically its software services. Third-party vendors will achieve innovation through the release of software services complementary to the core software services and bridge software services of platform providers.

VI. Conclusion

In this article, we compared the positions of software services of a platform provider in a software service network with those of the third-party vendors. For the analysis, we used empirical data gathered from the AppExchange of Salesforce.com. The analysis results showed that the software services released by the platform provider locate at a central position in the entire network, at the core positions in service clusters, and at the bridge positions between the clusters, while software services released by the third-party vendors were complementary to those software services of the platform provider. A central position in the entire network meant that software services at this position lead the entire software service ecosystem. A software service goes to a central position as more platform users use the software service. A software service linked to the software services of other clusters meant that it bridges service clusters, allowing new knowledge to flow into the clusters.

The novelty of our findings is that it demonstrats that the platform provider does not just provide a platform, on which the third-party vendors can innovate, but also participates in the innovation process itself. Moreover, the role of the platform provider is to lead the innovation process within the ecosystem. The platform provider needs to motivate third-party vendors to innovate through the provisioning of complementary services near to the core and bridge services that the platform provider made available. This was an aspect, which prior research had missed and what decision makers would likely miss without this research. This understanding will help platform providers to design strategies (e.g., based on incentives) for making third-party vendors create valuable software services on their platforms. Therefore, our findings are expected to steer research on innovation from a pure analysis of the behavior of platform users (collective intelligence) in networks toward the analysis of the roles and activities of platform providers.

Acknowledgment

The authors would like to thank the reviewers of this article, who gave highly valuable feedback for improving our research.

References

[1] P. Levy, "From social computing to reflexive collective intelligence: The IEML research program," *Inf. Sci.*, vol. 180, no. 1, pp. 71–94, Jan. 2010.

[2] A. Weiss, "The power of collective intelligence," *NetWorker*, vol. 9, no. 3, pp. 16–23, Sep. 2005.

[3] H. W. Chesbrough, *Open Innovation: The New Imperative for Creating and Profiting from Technology*, 1st Trade Paper ed. Boston, MA, USA: Harvard Bus. School Press, 2005.

[4] J. Fuller, K. Hutter, J. Hautz, and K. Matzler, "User roles and contributions in innovation-contest communities," *J. Manage. Inf. Syst.*, vol. 31, no. 1, pp. 273–308, Summer 2014.

[5] S. Saonee, A. Manju, S. Suprateek, and S. Kirkeby, "The role of communication and trust in global virtual teams: A social network perspective," *J. Manage. Inf. Syst.*, vol. 28, no. 1, pp. 273–310, Summer 2011.

[6] E. von Hippel, "Innovation by user communities: Learning from open-source software," *MIT Sloan Manage. Rev.*, vol. 42, no. 4, pp. 82–86, 2001.

[7] A. Gawer and M. A. Cusumano, *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*, 1st ed. Brighton, MA, USA: Harvard Bus. School Press, 2002.

[8] K. J. Boudreau, "Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation," *Org. Sci.*, vol. 23, no. 5, pp. 1409–1427, Sep. 2011.

[9] C. Y. Baldwin and C. J. Woodard, "The architecture of platforms: A unified view," in *Platforms, Markets and Innovation*. Cheltenham, U.K.: Edward Elgar Publishing, 2009.

[10] A. Gawer and M. A. Cusumano, "How companies become platform leaders," *MIT Sloan Manage. Rev.*, vol. 49, no. 2, pp. 28–35, Winter 2008.

[11] A. MacCormack, C. Baldwin, and J. Rusnak, "Exploring the duality between product and organizational architectures: A test of the 'mirroring' hypothesis," *Res. Policy*, vol. 41, no. 8, pp. 1309–1324, Apr. 2008.

[12] M. Cusumano, "Cloud computing and SaaS as new computing platforms," *Commun. ACM*, vol. 53, no. 4, pp. 27–29, Apr. 2010.

[13] T. R. Eisenmann, G. Parker, and M. van Alstyne, "Strategies for two-sided markets," *Harvard Bus. Rev.*, vol. 84, pp. 92–101, Oct. 2006.

[14] E. G. Weyl, "A price theory of multi-sided platforms," *Amer. Econ. Rev.*, vol. 100, no. 4, pp. 1642–1672, 2010.

[15] G. Parker and M. van Alstyne, "Innovation, openness, and platform control," *Management Sci.*, vol. 64, no. 7, pp. 3015–3032, Jul. 2018.

[16] A. Ghazawneh and O. Henfridsson, "Balancing platform control and external contribution in third-party development: The boundary resources model," *Inf. Syst. J.*, vol. 23, no. 2, pp. 173–192, Mar. 2013.

[17] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—The business perspective," *Decis. Support Syst.*, vol. 51, no. 1, pp. 176–189, 2011.

[18] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Netw.*, vol. 1, no. 3, pp. 215–239, 1978/1979.

[19] A. Tiwana, B. Konsynski, and A. A. Bush, "Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics," *Inf. Syst. Res.*, vol. 21, no. 4, pp. 675–687, Dec. 2010.

[20] C. Y. Baldwin and C. J. Woodard, "Platforms, markets and innovation," in *The Architecture of Platforms: A Unified View*, A. Gawer, Ed. Edward Elgar Publishing, 2009.

[21] C. Y. Baldwin and K. B. Clark, *Design Rules: The Power of Modularity*, vol. 1. Cambridge, MA, USA: MIT Press, 1999.

[22] B. Iyer, C.-H. Lee, and N. Venkatraman, "Managing in a 'small world ecosystem': Lessons from the software sector," *Calif. Manage. Rev.*, vol. 48, no. 3, pp. 28–47, 2006.

[23] J. Wareham, P. B. Fox, and J. L. Cano Giner, "Technology ecosystem governance," *Org. Sci.*, vol. 25, no. 4, pp. 1195–1215, Aug. 2014.
[24] A. Gawer, *Platforms, Markets and Innovation*. Cheltenham, U.K.: Edward Elgar Publishing, 2009.
[25] B. Bergvall-Kåreborn and D. Howcroft, "Persistent problems and practices in information systems development: A study of mobile applications development and distribution," *Inf. Syst. J*, vol. 24, no. 5, pp. 425–444, Sep. 2014.
[26] K. Karhu, A. Botero, S. Vihavainen, T. Tang, and M. Hämäläinen, "A digital ecosystem for co-creating business with people," *J. Emerg. Technol. Web Intell.*, vol. 3, no. 3, pp. 197–205, 2011.
[27] K. Boudreau, "Open platform strategies and innovation: Granting access vs. devolving control," *Manage. Sci.*, vol. 56, no. 10, pp. 1849–1872, Sep. 2010.
[28] H. Chesbrough, *Open Innovation: Researching a New Paradigm*. Oxford, U.K.: Oxford Univ. Press, 2008.
[29] M. Cusumano, "Technology strategy and management: The evolution of platform thinking," *Commun. ACM*, vol. 53, no. 1, pp. 32–34, Jan. 2010.
[30] J.-C. Rochet and J. Tirole, "Platform competition in two-sided markets," *J. Eur. Econ. Assoc.*, vol. 1, no. 4, pp. 990–1029, Jun. 2003.
[31] C. A. Hidalgo, B. Klinger, A.-L. Barabási, and R. Hausmann, "The product space conditions the development of nations," *Science*, vol. 317, no. 5837, pp. 482–487, 2007.
[32] R. Albert, H. Jeong, and A.-L. Barabasi, "Internet: Diameter of the world-wide web," *Nature*, vol. 401, no. 6749, pp. 130–131, Sep. 1999.
[33] K. Kim, W.-R. Lee, and J. Altmann, "SNA-based innovation trend analysis in software service networks," *Electron. Markets*, vol. 25, pp. 61–72, Mar. 2015.
[34] K. Kim and J. Altmann, "Evolution of the software-as-a-service innovation system through collective intelligence," *Int. J. Cooperative Inf. Syst.*, vol. 22, no. 3, 2013, Art. no. 1340006.
[35] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.
[36] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Nov. 2002.
[37] J. Scott, *Social Network Analysis: A Handbook*, 2nd ed. Thousands Oaks, CA, USA: Sage, 2000.
[38] K. Kim and J. Altmann, "Effect of homophily on network formation," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 44, pp. 482–494, 2017.
[39] L. Festinger, K. W. Back, and S. Schachter, *Social Pressures in Informal Groups: A Study of Human Factors in Housing*. Stanford, CA, USA: Stanford Univ. Press, 1950.
[40] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, 2001.
[41] T. C. Schelling, *Micromotives and Macrobehavior*, Revised ed. New York, NY, USA: W.W. Norton & Company, 2006.
[42] R. Burt, *Structural Holes: The Social Structure of Competition*. Cambridge, MA, USA: Harvard Univ. Press, 1995.
[43] A. B. Hargadon, "Brokering knowledge: Linking learning and innovation," *Res. Org. Behav.*, vol. 24, pp. 41–85, 2002.
[44] G. Palla, A.-L. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, Apr. 2007.
[45] T. S. Kuhn, *The Structure of Scientific Revolutions*, 3rd ed. Chicago, IL, USA: Univ. Chicago Press, 1996.
[46] M. S. Granovetter, "The strength of weak ties," *Amer. J. Sociol.*, vol. 78, no. 6, pp. 1360–1380, May 1973.
[47] R. S. Burt, "Structural holes and good ideas," *Amer. J. Sociol.*, vol. 110, no. 2, pp. 349–399, Sep. 2004.
[48] K. Kim, J. Altmann, and J. Hwang, "An analysis of the openness of the web2.0 service network based on two sets of indices for measuring the impact of service ownership," in *Proc. 2011 44th Hawaii Int. Conf. Syst. Sci.*, Kauai, HI, USA, 2011, pp. 1–11.
[49] D. Krackhardt and R. N. Stern, "Informal networks and organizational crises: An experimental simulation," *Social Psychol. Quart.*, vol. 51, no. 2, pp. 123–140, 1988.
[50] B. Nooteboom, W. van Haverbeke, G. Duysters, V. Gilsing, and A. van den Oord, "Optimal cognitive distance and absorptive capacity," *Res. Policy*, vol. 36, no. 7, pp. 1016–1034, Sep. 2007.
[51] R. Qu and Q. Ye, "Service ecosystems of cloud computing," *Proc. 1st Int. Conf. E-Business Intell. (ICEBI)*, 2010.
[52] S. Jansen and S. Brinkkemper, *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry*. Northampton, MA, USA: Edward Elgar Publishing, 2013.
[53] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.
[54] R. Guimerà and L. A. N. Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, Feb. 2005.
[55] T. G. Lewis, *Network Science: Theory and Applications*, 1st ed. Hoboken, NJ, USA: Wiley, 2009.
[56] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, Dec. 2009.
[57] A.-L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.
[58] W. H. Greene, *Econometric Analysis*, 7th ed. Boston, MA, USA: Prentice-Hall, 2011.
[59] J. D. Gibbons and S. Chakraborti, *Nonparametric Statistical Inference*, 5th ed. Boca Raton, FL, USA: Chapman and Hall/CRC Press, 2010.
[60] I. Olkin, Ed., *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, 1st printing ed. Stanford, CA, USA: Stanford Univ. Press, 1960.
[61] C. S. Wagner and L. Leydesdorff, "Network structure, self-organization, and the growth of international collaboration in science," *Res. Policy*, vol. 34, no. 10, pp. 1608–1618, Dec. 2005.
[62] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, New York, NY, USA, 1999, pp. 251–262.
[63] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 2, pp. 404–409, Jan. 2001.
[64] T. Eisenmann, G. Parker, and M. van Alstyne, "Opening platforms: How, when and why?" in *Platforms, Markets and Innovation*. London, U.K.: Edward Elgar Publishing, 2009, pp. 131–162.
[65] T. Eisenmann, G. Parker, and M. van Alstyne, "Platform envelopment," *Strat. Manage. J.*, vol. 32, no. 12, pp. 1270–1285, Dec. 2011.
[66] N. Haile and J. Altmann, "Structural analysis of value creation in software service platforms," *Electron. Markets*, vol. 26, no. 2, pp. 129–142, May 2016.
[67] N. Haile and J. Altmann, "Estimating the value obtained from using a software service platform," in *Economics of Grids, Clouds, Systems, and Services*. New York, NY, USA: Springer, 2013, pp. 244–255.
[68] N. Haile and J. Altmann, "Evaluating investments in portability and interoperability between software service platforms," *Future Gener. Comput. Syst.*, vol. 78, no. 1, pp. 224–241, 2018.
[69] K. Kim, J. Altmann, and W. Kim, "On the time lag of the effect of network position on service performance in software service networks," *Inf. Manage.*, 2019. [Online]. Available: https://doi.org/10.1016/j.im.2019.02.004
[70] N. Haile and J. Altmann, "Value creation in software service platforms," *Future Gener. Comput. Syst.*, vol. 55, pp. 495–509, 2016.
[71] S. Beak, K. Kim, and J. Altmann, "Role of Platform Providers in Service Networks: The Case of Salesforce. com App Exchange," in *CBI 2014, 16th IEEE Conf. Business Inf.*, Geneva, Switzerland, 2014.