The Co-Evolution of Digital Ecosystems

Completed Research Paper

SungYong Um
Department of Information Systems
School of Computing
National University of Singapore
13 Computing Drive, 117417, Singapore
sungyong @nus.edu.sg

Youngjin Yoo
Weatherhead School of Management
Case Western Reserve University
10900 Euclid Avenue, Cleveland, Ohio, 44106, USA
youngjin@case.edu

Abstract

Digital ecosystems continue to evolve, as new external APIs continue to enter into it. Not all new APIs, however, have the same fate: some are successfully connected with existing APIs and spark major changes in the ecosystem, while others are simply ignored. Particularly, some of the components end up playing a major role in shaping the structure of the ecosystem. To systematically explore what determines the fate of different APIs, we hypothesize how the network property and non-network property of external APIs affect the probability of the APIs becoming a part of the core in the structure of a digital ecosystem. For the empirical test, we used plug-in source code data collected from Wordpress.org from January 2004 to December 2014. Using a survival analysis, we found that external APIs as cores are more influential than components offered by a focal platform system in the growth of a digital ecosystem.

Keywords: *a digital ecosystem, digital platform system, digital innovation, heterogeneous API, add-on product*

Introduction

The dynamic generative growth of a digital ecosystem is what makes digital innovation unique. Digital platforms increase its functional scope and scale by leveraging complementary add-on products from third-party developers in their platform system (Cusumano and Gawer 2002; Eisenmann et al. 2006; Parker and Van Alstyne 2005). A digital ecosystem consists of a focal platform, a large number of heterogeneous complementary add-on products, and a host of boundary resources that are used to connect the focal platform and the complementary products (Ghazawneh and Henfridsson 2012). Application Programming Interfaces (API) are important boundary resources that allow third-party developers to create add-on products for the focal platform (Eaton et al. 2015). In a digital ecosystem, APIs act as fundamental component in creating complementary digital products as developers often stitch a number of APIs to create a new add-on product. Prior studies on digital ecosystems reveal three important aspects of APIs as digital components in a digital ecosystem (Um et al. 2013). First, not all APIs that are used in the platform are created by the platform owner. For example, for the WordPress ecosystem, there are 443 APIs used in the ecosystem as of December 2012; among them, only 103 of them are created by the platform owner. We refer those APIs that are created by the platform owners as *internal* APIs, while those that are created by other companies as external APIs. Second, the interaction among APIs produces the landscape of the ecosystem. Interaction among APIs refers to the pattern by which certain APIs are co-used by different complementary products. Finally, new components continue enter into the ecosystem, which in turn changes the boundary of the ecosystem's landscape (Yoo et al.

2010). However, we know little about how certain components becomes core components that re-shape the structure of a digital ecosystem, while other components remain insignificant.

When new APIs are introduced into a digital ecosystem, they can diversify the given ecosystem potentially in unexpected ways as they interact with existing APIs. As such, a digital ecosystem co-evolves with APIs that make up the ecosystem. From an API designer's perspective, it is strategically important that their APIs become a part of core components in the ecosystem that are highly connected with other components, as they can gain economic benefits and create a large user install base for their own service offering when their APIs become a part of core of a popular ecosystem. Thus, the evolution of a digital ecosystem includes not only the functional diversity of a platform system through the emergence of new complementary add-on products, it also represents the growth in the scale and scope of APIs that are used to create those digital products. Past research on digital ecosystems focuses on the role of platform owners and their provision of APIs as boundary resources. However, given the large number of external APIs used in digital ecosystems, it is crucial to understand how the introduction of new external APIs affects the evolution of an ecosystem. That's what we are studying in this paper.

Specifically, we focus on an evolutionary network perspective to study how the connection patterns among APIs change over time as new external APIs enter into the ecosystem. An earlier study (Um et al. 2013) shows that complementary add-on products in an ecosystem can be clustered based on their use of APIs. In this study, we explore how new APIs affect the existing structure of clusters of add-on products in a digital ecosystem. In particular, since not all new APIs lead to structural changes in the ecosystem, we are trying to understand under what conditions new external APIs lead to the structural change of the ecosystem. Specifically, this study asks:

> Under what conditions, do new external APIs create structural diversification of an existing digital ecosystem?

To explore the role of new APIs on the structural change of a digital ecosystem, we first examine the evolution of an ecosystem over time. To do so, we created a network of add-on products and measure topological overlap among add-on products to identify the clusters of digital products based on the similar usages of APIs for each time period. Given the structure of a digital ecosystem based on a digital product network at a given point in time, we then built a network of APIs for the same time period to understand how the changes in APIs affect the structure of a digital ecosystem. We examine the relationship between the two networks over time by exploring the role of network and non-network properties on the structural change of a digital ecosystem to effectively capture the dynamic growth pattern.

We use data from the WordPress ecosystem. WordPress is the world's largest blog ecosystem, offering a diverse set of functions from a large number of add-on digital products (plug-ins). Third-party developers combine APIs offered by WordPress and other platform systems such as Google or Facebook to construct new plug-ins. We used the data of WordPress from its conception until December 2014.

The remainder of this paper is organized as follows. First, we review existing literature on digital ecosystem dynamics. Second, we explain the network dynamics driven by the introduction of new APIs. Third, we describe the empirical model and results of our analysis. Finally, we discuss the theoretical and methodological implications of this study.

Literature Review

Characteristics in a digital ecosystem

An ecosystem is built to extend the functionality of a platform by mobilizing third-party developers (Gawer and Cusumano 2014). A platform is modularized into a set of decomposable components in order to simplify the complex interactions of components in a system (Sosa et al. 2004; Ulrich 1995). The third party developers design and offer their products following the design rules controlled by the owner of a platform system (Baldwin and Clark 2000; Sanchez and Mahoney 1996; Tushman and Anderson 1986; Tushman and Rosenkopf 1992). In this scenario, an ecosystem expands the functional variety by enhancing the functionality of the products in a way that the platform owner alone cannot offer effectively

(Eisenmann et al. 2011). As the functional variety of products offered in an ecosystem grows, its structure is continuously differentiated.

A digital ecosystem is generative compared to a traditional one (Yoo et al. 2012), as it does not have a pre-defined single set of design rules tightly controlled by a digital platform owner (Yoo et al. 2010). The core functionality of platforms' boundary resources is open to the public so that other heterogeneous digital resources from other platforms can be used to create novel and heterogeneous add-on products that often go far beyond the original design vision of the platform owner (Yoo 2013). Specifically, third-party developers do not have to narrowly following design rules under the control within a single design hierarchy (Boland et al. 2007; Yoo et al. 2012). Therefore, a digital ecosystem makes the existing boundary structure of a digital platform system be more complex, unbounded and dynamic over time and enables the emergence of multiple hierarchies over time.

Structure in a digital ecosystem

Following an earlier study (Um et al. 2013), we conceptualize a digital ecosystem with a multi-layered hierarchical structure as shown in Figure 1. The first layer is composed of complementary add-on products, while the second layer consists of digital components, specifically APIs. Add-on products are created by combining APIs, forming hierarchical clusters based on their patterns of API usages. Over time, new clusters of add-on product layer emerge as the ecosystem grows in size and scale. The diversification and growth of the add-on product layer represents the functional heterogeneity of the ecosystem.

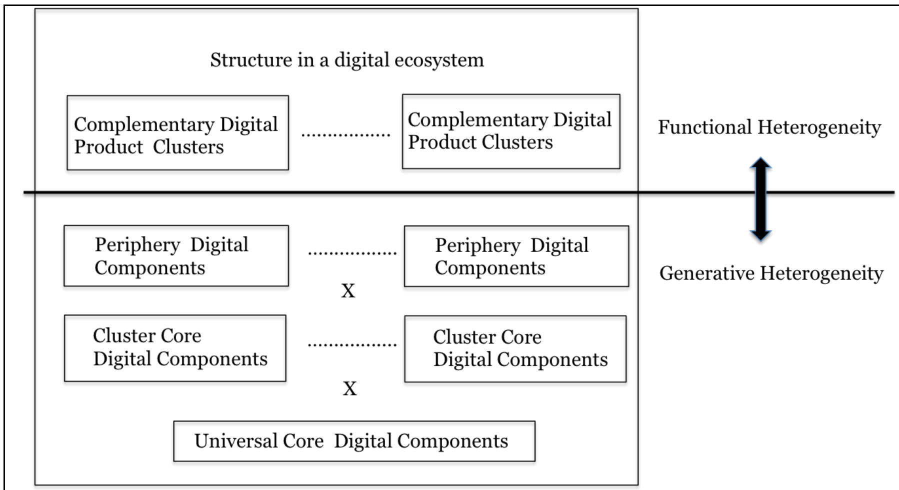

Figure 1. The structure of a digital ecosystem

The cluster structure of the add-on product layer is supported by a wide range of heterogeneous APIs that are used in creating those add-on products. Not all APIs play equally important roles in shaping the way add-on products are created. As shown in Figure 1, the component layer can be further divided into three layers (Um et al. 2013): universal core APIs, cluster core APIs, and periphery APIs (Csermely et al. 2013; Rombach et al. 2014). Universal core APIs are a few internal APIs that are ubiquitous among all add-on products across all clusters in the ecosystem. The extent of usage of these is higher than that of most other APIs. They are densely connected with other APIs, forming the foundation the whole ecosystem. Cluster core APIs are a few internal and external APIs that play important roles in shaping each cluster. They are used to create *the functional heterogeneity across different clusters*. The frequency of usage of these APIs is higher than for other digital components except universal digital components in each cluster. Periphery APIs are used to increase *the functional variety within each cluster*[1].

Theory Development

Co-evolutionary dynamics in a digital ecosystem

This study adopts the evolutionary network perspective to understand the changes in the structure of a digital ecosystem over time when new APIs are introduced. In general, the evolutionary model represents the selection, variation, and retention processes in an ecosystem (Agarwal et al. 2004). In a digital ecosystem characterized by the open and unbounded boundaries, functional heterogeneity arises from the introduction of new APIs. New combinatorial pattern caused by the new APIs can lead to the emergence of new add-on product categories, if there are enough separation between the new types of add-on products that use new APIs, compared to the existing add-on products.

We focus specifically on the microscopic evolutionary concepts of *migration* and *mutation* to effectively explore different roles of APIs in the emergence of new clusters (Bornholdt and Sneppen 1998; Draghi et al. 2010; Lacy 1987; Wagner and Altenberg 1996; Xia and Levitt 2002). Migration represents the exogenous interbreeding process of the API variations that ultimately leads to the selection of functional varieties of add-on products (Woodard 2008). As such, a rich and infinitely increasing set of foreign APIs possibly promotes the structural diversification within and across clusters of add-on products in an ecosystem. Mutation, on the other hand, represents the continuously changes of a cluster as new APIs continue to migrate into the cluster that ultimately leads to the emergence of new clusters (Lynch 2002; Lynch 2010; Ohta and Kimura 1969). New APIs allows the deletion, duplication, or substitution of the subsets of existing combination of APIs (Spears 1992; Wagner 2003; Wagner et al. 2007). Thus, the migration of new APIs might cause mutation of digital products and their clusters.

Thus, migration and mutation are two fundamental mechanisms of co-evolutionary dynamics in a digital ecosystem where new APIs and new add-on products continue to emerge over time, increasing functional heterogeneity of the ecosystem (Baldwin and Clark 2000; Yoo et al. 2012). The microscopic evolutionary perspective allows us to systematically explore the co-evolution of a digital ecosystem.

The simultaneous dynamics of two different networks

In this study, we focus on the central role of external APIs in the co-evolution of a digital ecosystem. We focus on the conditions when the introduction of new external APIs lead to changes in the interaction patterns among APIs in such a way that lead to the fundamental changes in the underlying cluster structures of add-on products. The owner of a platform system has limited control over how third-party developers choose to use external APIs. In a digital ecosystem, there is an inherent competition among external APIs to play more central role in the ecosystem. For external API providers, the competition is mainly to extend the reach of their services across different ecosystems. For example, Google wants its Google Map APIs to be used in as many digital ecosystems (WordPress, iOS, Android, etc) as possible in competition with other similar map services. Furthermore, API providers want to make their APIs more central in the ecosystem by making them attractive to many third-party developers. Thus, when a new API first emerges, the API designer want existing third-party developers to incorporate the new API into the existing add-on products.

To study how the migration of external APIs affects the mutation of add-on product clusters, we examine simultaneously two different networks (APIs and add-on products). First, in an add-on product network, a node indicates an add-on product, and an edge represent the link between two add-on products that share the same API. In this network, add-on products form clusters depending on their similar combinatorial uses of APIs which form the basis of the structure of a digital ecosystem as described in

[1] We use the term “functional heterogeneity” to denote differences of different kind that are found across add-on products clusters, while “functional variety” to denote differences of the same kind that are found within a cluster.

Figure 1. A cluster will be split if the migration of APIs causes the change of the network pattern (that is, how different add-on products share APIs).

Second, in an API network, a node represents an API, and an edge indicates that the two APIs are used for the same add-on products. With an API network, we can explore how much new APIs are diversely and strongly interconnected with others. As APIs and add-on products grow, the number of nodes and edges of an API network continues to grow, the pattern of which is quite hard to predict. A functionally useful APIs can continuously generate new diverse connections with other APIs in various ways. After new APIs migrate into a digital ecosystem, their connection with other APIs grows in different ways. Some APIs start to frequently connected with small numbers while the others are widely interact with the multiple APIs with less frequently. Thus, the extent of connection patterns over time dynamically affects the layered structure. Such a change can be a key factor that causes a structural change of a digital ecosystem such as an emergence of a new cluster or changes with a cluster of add-on products.

New external APIs are continuously introduced into a digital ecosystem, and the extent of their usage changes over time. Particularly, as the number of APIs increases, the connection among APIs leads to the changes in cluster structures in a digital ecosystem. More specifically, if the connections of new APIs follow existing combinatorial pattern among APIs in the cluster, it will lead to a within-cluster differentiation. A within-cluster differentiation occurs when new APIs are introduced just to modify the function of existing add-on products. A within-cluster differentiation thus implies that a sub-division of an existing cluster occurs. If the connections of new APIs show new combinatorial patterns, however, it is likely to suggest new types of add-on products are being introduced into the ecosystem. If new types of add-on products survive, ultimately, they are likely to lead to the emergence of a completely new cluster.

Figure 2 presents a hypothetical example of the co-evolutionary dynamics driven by a new external API that becomes a cluster core API over time. An API network is shown on the left-hand side, and the structure of an add-on products network is shown on the right-hand side. The connection between the two networks shows the co-evolutionary dynamics between APIs and add-on products.

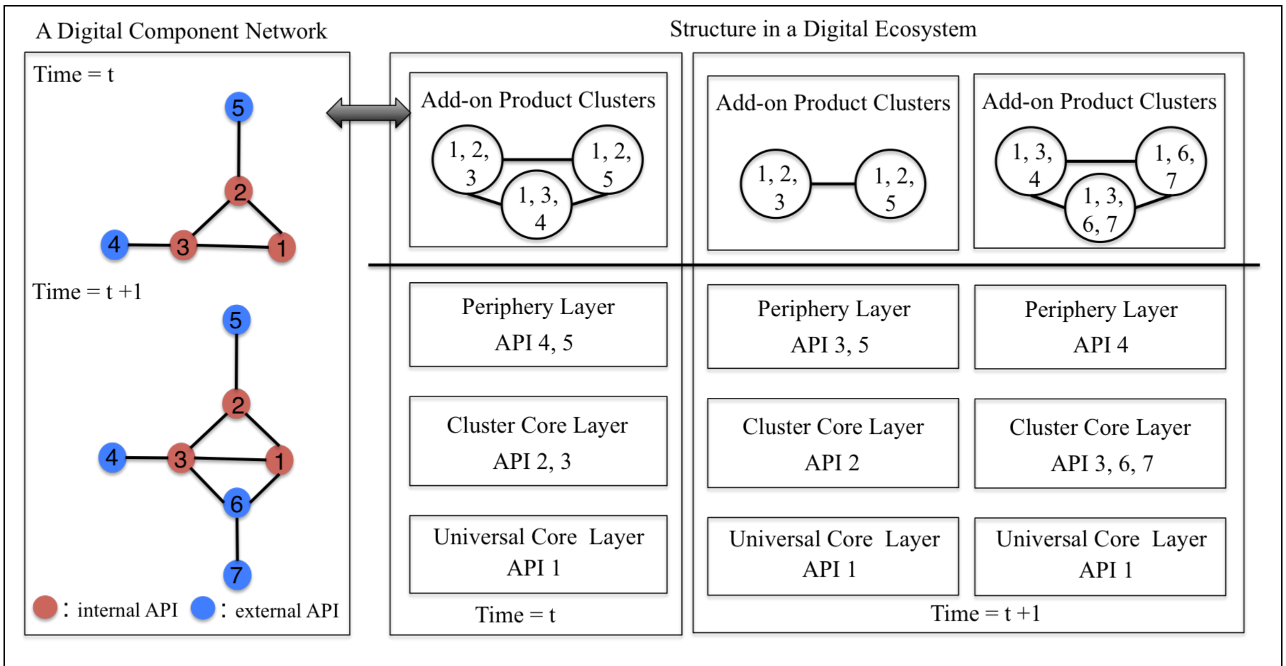

Figure 2. Co-evolutionary dynamics of a digital ecosystem

In this hypothetical example, at time t, there are three internal APIs and two external APIs that are used to build three add-on products all in a single cluster in the ecosystem on the right-hand side. Here, API 1 is used by every add-on and is a universal core API. API 2 and 3 are used to form the cluster among the three add-on products, thus forming cluster core APIs[2]. API 4 and 5 are combined with other APIs to increase the functional variety within the cluster. As such, they are understood as periphery APIs. Based on the use in the add-on products, those five APIs form an API network at time t on the left-hand side. Internal API 1, 2, and 3 constructed the strong connection, as they actively contributed to build the add-on products. External API 4 and 5 remained to differentiate the functional diversity in the API network at time t.

At time t+1, two new external API 6 and 7 (color blue nodes) migrate into a digital ecosystem. Those APIs are used to create two new add-on products that make the add-on product cluster to mutate into the two clusters. The right-hand side of clusters at time t+1 show two new add-on products, which led the structure to be mutated. In this example, the external API 6 and 7 popularly used and became a part of cluster core of new cluster, together with API 3. In this example, one of the add-on products from the old cluster now resembles more the new two add-on products, thus separated from the original cluster, which now has only 2 add-on products in the cluster. The hypothetical example on the left-hand side shows that external API 6 gains relative high degree and frequency in the API network compared to API 7 at time t+1. The use of API 6 enables to move into the center of the network. However, though API 7 is included in the cluster core layer the same with API 6, the limited degree of interaction with other APIs makes it not to be included in the center of a network. This hypothetical example is to show the basic mechanism of co-evolution of a digital ecosystem through migration of new APIs and mutation of add-on products and their clusters over time.

Hypotheses

External APIs in a digital ecosystem play important roles as they create unexpected, and open-ended innovations in the ecosystem. The role of external APIs is not pre-determined, so they can be combined with other APIs depending on their functional usefulness (Yoo et al. 2012). Thus, their connection with other APIs continue to mutate that produces unexpected interaction patterns (Arthur 2009). In such dynamic interactions, the pattern of connectivity is not just dominated by few APIs which are popular in their uses or systematically critical to be used. Depending on their functional features, external APIs can show a broad range of connectivity that can ultimately change the existing structure. We specifically seek to understand the attributes of external APIs that affect the changes in the pattern of connectivity in terms of degree and frequency in an API network (Watts and Strogatz 1998). The two characteristics of nodes—degree and frequency—are separately used to differentiate the role of an API that increase either within-cluster differentiation or the emergence of new clusters.

Each API is connected with other APIs, as APIs are used to crease add-on products by third party developers. The degree of the API in the API network grows. The degree indicates the total number of nodes connected to a focal node. Thus, the degree implies the extent to which an API is connected with other APIs. A high-degree external APIs is connected with a broad range of other APIs, forming a building block that shapes the behavior the ecosystem. Furthermore, as the size of the building block grows, the influence of the external API increases. These external APIs with high degree can affect a critical mass of other APIs enough to re-order the existing pattern of connectivity by leading the evolution of the existing pattern of connectivity, as any APIs in the building block can more easily create new connections with other previously unconnected APIs than APIs outside the building block. The wide usages of external API in an API network lead to the increased usages on the generation of new add-on products across clusters in an add-on product network. Therefore, external APIs with high degree have a high chance of becoming APIs across newly emerged clusters in an add-on product network. Thus, we posit:

H1*: External APIs with a high degree have a higher probability of being APIs of newly emerged clusters of add-on products in a digital ecosystem.*

[2] To simplify the hypothetical example in Figure 2, we only show a single cluster in time t. However, one can assume that there are other clusters in time t where other APIs not shown in this example form cluster core APIs of other clusters.

In an API network, some APIs are more frequently connected with other APIs than others. Frequency represents the number of connections between two APIs. As such, frequency implies the weighted value of an edge. External APIs with a high frequency diversify the existing connection patterns on their own through reshaping the existing hierarchical structure as they are successfully connected with multiple internal APIs. Those external APIs are likely to separate a single cluster built by internal components into multiple clusters. Therefore, the more frequently external APIs connect with others, the more diverse the existing structure becomes. External APIs having a high frequency are likely to contribute to the generation of functionally distinct add-on products that forms their own clusters in a digital ecosystem. Those external APIs show high usages in each cluster. Thus, we hypothesize:

H2*: External APIs with a high frequency have a higher probability of being APIs of newly emerged clusters of add-on products in a digital ecosystem.*

External APIs are offered from other platform providers, and their functional types are more varied compared to those of internal APIs. Third-party developers have limitations to access the broad range of functional information in a short period of time. However, once external APIs are created and introduced to a digital ecosystem for a while, third-party developers would have a chance to observe how other developers are using new external APIs and can mimic their design, particularly the add-ons using those APIs gain popularity among users (Johnson et al. 2014). Therefore, older external APIs have a higher chance than younger ones to increase their connectivity with other APIs, since they have more chances of being exposed to wider range of third-party developers. This then, in turn, lead to a higher probability of them to become APIs in the newly emerged clusters. Thus, we hypothesize:

H3: *Older external APIs have a higher probability of being APIs of newly emerged clusters of add-on products in a digital ecosystem.*

A digital ecosystem does not have a specific functional boundary. Any external APIs from various platforms can contribute to the growth of a digital ecosystem. Some APIs come from large, reputable firms such as Google and Yahoo, while others do not. Third-party developers can easily learn about these large firms and adopt their APIs. As such, external APIs offered by large providers that generate a large number of APIs have a high chance of migrating into a digital ecosystem and of connecting with other APIs. By contrast, external APIs by small providers that produce only few APIs have limited opportunities to be noted by third-party developers. Though these components may be useful, they have a low probability to be recognized and used by third-party developers. Furthermore, the number of APIs in a digital ecosystem is exponentially increasing over time and online communities have limited time to share information about them (Johnson et al. 2014). Therefore, even though external APIs might have useful functionality, if they are not by large providers that generate many APIs, they will have a limited chance of contributing to the emergence of clusters in a digital ecosystem. Thus, we hypothesize:

H4: *External APIs by firms that produce a large number of APIs have a higher probability of being APIs of newly emerged clusters of add-on products in a digital ecosystem.*

Analytical Approach

This paper mainly uses two analytical approaches. First, to measure the structural mutation of a digital ecosystem, we focus on the changes in clusters of add-on products. We generated clusters using co-expression network analysis. Second, in order to identify the impact of migration of external APIs on the structural mutation, we use a survival model to assess the likelihood of the emergence of new clusters using the extracted API information from the API network.

Measuring the evolution of add-on products clusters structure

We used co-expression network analysis (Horvath 2011) to capture the cluster dynamics that take place in a digital ecosystem. This analysis mainly captures the topological overlap of add-on products (Ravasz and Barabási 2003; Ravasz et al. 2002) based on the extent of similar uses of APIs. It mainly considers the number of APIs that add-on products share and the number of times APIs are repetitively used among add-on products. A node indicates an add-on product, and an edge represents the weighted value of commonly shared number of APIs between a pair of add-on products (Horvath and Dong 2008). The weighted value is measured by the topological overlap of APIs' recombinatorial interactions among add-on products that construct the underlying structure of clusters (Ravasz et al. 2002). First, An adjacency matrix is estimated between plug-ins *i* and *j* indicated as a_{ij}. The matrix displays the weighted entries of N by N add-on products that range from 0 to 1. a_{ij} is equal to 1 when add-on products *i* and *j* commonly include the same set of APIs. If an add-on product does not share the same APIs, a_{ij} will be displayed as 0. Second, we apply the topological overlap measure to estimate the relative inter-connectedness among add-on products (Doncheva et al. 2012; Ravasz et al. 2002). The converted entries of topological overlap matrix from the adjacency matrix indicate the weighted similarity measure in an add-on product network (Zhao et al. 2010). The main goal of getting a topological overlap measure is to capture the hierarchical cluster structure in a digital ecosystem in a statistically robust way and its mutation (split and merge) through the generation of sub-clusters as new add-on products with different combinatorial patterns of APIs are continuously created (Yip and Horvath 2007). Therefore, by capturing the generation of clusters, we can understand the structural dynamics in a digital ecosystem by visualizing the topological matrix. Specifically, it is calculated in the following way:

$$\text{Topological overlap measure} = \frac{\sum_{u \neq i,j} a_{iu} a_{ju} + a_{ij}}{\min(k_i + k_j) + 1 - a_{ij}},$$

where $k_i = \sum a_{ij}$ implies the correlation between an add-on product *i* and its neighbor add-on product *j* based on the common APIs used by them, and $\sum_{u \neq i,j} a_{iu} a_{ju}$ indicates the number of APIs commonly used between add-on products *i* and *j*.

A survival model to estimate the impact of API migration on mutation

The second approach is to understand the role of new external APIs on the generation of clusters in a digital ecosystem. We used a survival model to capture the role of new external APIs in periods of new events when new clusters are generated. The hazard function used in a survival model represents the rate at which events such as death or progression take place in each entity during the specific time period. Thus, the rate of risk indicates the relative probability of events happening compared to the control group, or time-invariant entities. In general, as the meaning of hazard indicates, an event is defined as the death of an entity. However, in the context of this study, an event is defined as the birth of new clusters, so that the interpretation of hazard needs to be the opposite. We used a Weibull hazard function for our panel data set because the ratio of hazards changes and the predictors show parametric values. Specifically, the Weibull model is displayed in the following way:

$$S(t) = \exp(-\lambda t^{p}),$$

where λ is a constant, t indicates time, and p implies the probability of an event occurring at time t.

A Survival model specification

The purpose of this study is to explore the impact of external APIs on the structural change of a digital ecosystem over time. The basic unit of analysis is the external API in a digital ecosystem where new external APIs are continuously introduced and used to generate new add-on products. Some external APIs cause existing clusters to mutate over time while others do not. We explore which APIs play a core role in making a structural change at a given time point. In particular, we focus on two different types of predictors to capture the core role of components in the change. First, we use the network property of external APIs in an API network: degree and frequency. Degree refers to the number of nodes connected to a node in a network; in this context, it is the range of connectivity, or how many other APIs are commonly used with a focal API. Thus, for the purpose of this paper, degree implies functional popularity. Frequency refers to how many times a node is connected with others and thus indicates the actual number an API is combined with others in creating add-on products. A large frequency means that an API contributes to the structural change, as the strong connection strength leads to structural change.

Therefore, degree and frequency are good network property of external APIs. Second, we used the non-network property of external APIs: API age and the size of APIs that an API provider offers. The API age is to explore if the duration of API stays in a digital ecosystem affects the emergence of new cluster, as the amount of time that an API is exposed to third party developers can determine the multiple uses of APIs. The size of APIs that an API provider offers is to ultimately understand if the functional heterogeneity is dependent on diverse API providers, as the exploration of diverse unique API function can be limited if a provider is not well known.

The survival model is specified in the following way.

$$\log h_i(t) = \alpha(t-1) + \beta_1 x_{1ij}(t-1) + \beta_2 x_{2ij}(t-1),$$

where x_{1ij} indicates network property, and x_{2ij} shows non-network property.

An Empirical Study

Data collection and processing

We collected data from WordPress. In this study, we focus on API information used in each plug-in to explore whether external APIs play certain roles in the change of a digital ecosystem landscape. An API indicates a digital component, and a plug-in represents an add-on product. We downloaded all the available source code information as of December 2014 for a total number of 23,895 plug-ins; the total amount of collected source codes is more than 100GB. We developed a text-mining program written in Java to extract information about the APIs used in the 23,895 plug-ins. Using the time stamps written in each source code, we could build a set of time series data that shows detailed information about which APIs were used in each plug-in from January 2004 to December 2014.

We first built plug-in x API monthly tables to get connectivity information about plug-ins. The entries are composed of binary numbers (0 and 1) to understand if an API is used (marked as 1) or not (marked as 0) in a plug-in. To effectively collect API information, the text-mining program automatically compared the source codes of all plug-ins with the list of more than 10,000 APIs collected from www.programmableweb.com as of December 2014 to capture the function call information used in each plug-in. From the text-mining process, we found that, in January 2004, there were 86 plug-ins using 44 APIs (40 internal APIs and 4 external APIs). In December 2014, there were 23,985 plug-ins using 443 APIs (113 internal APIs and 330 external APIs offered by 265 companies). The collected detailed information of plug-in x API data are used to capture the landscape change of a digital ecosystem.

After reprocessing the plug-in x API matrices, we built API x API monthly tables to capture API network information in order to explore how each API is directly connected with other APIs. In this data set, APIs are connected if they are used in a plug-in at the same time. In the API network, the core attributes include source and target nodes and their edge weight. A source node represents the starting point of connection, and a target node indicates the end point between two APIs. Edge weight implies how many times source and target nodes are connected. In this way, we could build 443 (the total number of APIs used in all plug-ins) by 3 (source node, target node, and edge weight) monthly matrices from January 2004 to December 2014. From these data, we extracted the age of APIs based on when they were introduced into the digital ecosystem. We can understand degree, frequency (or edge weight), core/periphery role (based on core/periphery analysis), platform company size (based on whether the APIs were offered by the same firm), API age (when the API is introduced in a digital ecosystem), and so on.

Results

Add-on products co-expression network and clusters of add-on products

The results of the analysis of cluster dynamics are provided in Figure 3. The X-axis and Y-axis represent plug-ins observed at each time point in the analyzed results. As such, the size of each axis increases over time. A color bar represents clearly segmented clusters of plug-ins showing the different combinatorial patterns of APIs based on hierarchical clustering technique (Langfelder et al. 2008). The topological overlap measure is the key to segment clusters, as it forms the hierarchical order of plug-ins depending on the uses of APIs. In particular, the hierarchical order of plug-in is calculated based on the result of the topological overlap measure to display the extent of similar API combination patterns among plug-ins. Thus, plug-ins on the top of the hierarchical order includes APIs more likely to use in common, while plug-ins including APIs less likely to repetitively used in common are located on the bottom.

In particular, in the region below a color bar, red color shows high similarity among plug-ins in each square, while yellow color shows low similarity. Plug-ins are color-coded in different density based on the topological overlap measure. From the analyses, we understood that each square in a diagonal region represents plug-ins that are grouped together depending on the common pattern of APIs used in each plug-in. The size of each square in a diagonal region indicates the number of plug-ins clustered together. Each square continuously grows and mutates. The length of each segment in a color bar matches the size of cluster. From the analysis, we could graphically observe how the structure of a digital ecosystem changes over time and statistically extract information about how APIs are used in each cluster.

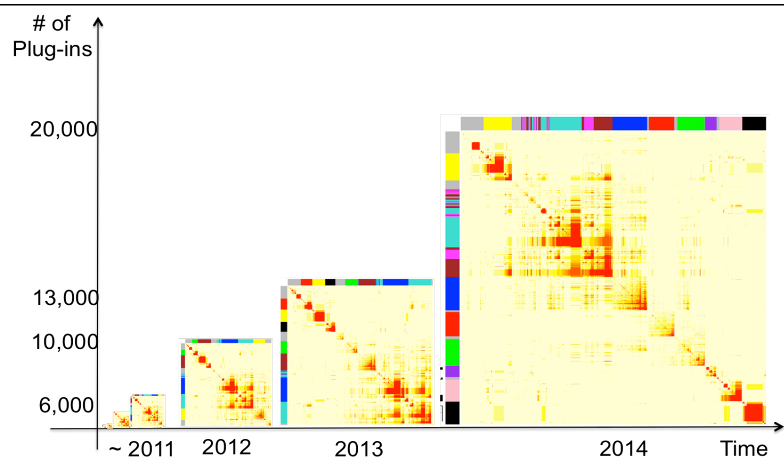

Figure 3. The structural dynamics of a digital ecosystem from 2004 to 2014

Statistical analysis

To specifically understand the role of external APIs, we then conducted a statistical analysis. The emergence of new clusters in the plug-in network in each time point is the dependent variable of this study. For the independent variables, we used the network property of API network. Variables such as degree and frequency are extracted through API network analysis. We also used the non-network property of APIs. Variables such as number of APIs a provider offering and API age are extracted through the log file of plug-in source codes.

We use different time points for the explanatory variables (time t-1) and the dependent variable (time t) because of the difference between when an API is introduced and when it is use, as third-party developers need time to discover them. We use binary values (0 and 1) for the change. Thus, the dependent variable displays whether the emergence of new cluster is taken place or not in each month. The following table shows the descriptive statistics of variables.

Variable	Obs	Mean	Std. Dev.	Min	Max
1. Degree of external APIs	45,064	4.955	13.695	0	169
2. Frequency of external APIs	45,064	2.309	13.119	0	457
3. API age	45,064	15.624	115.988	0	132
4. Number of APIs a provider offering	45,064	3.922	8.664	0	33

Table 1. Descriptive statistics

From the survival model, we can get the chance of being in the cluster core across clusters occurring in the given time period. The chance of change is indicated as a hazard ratio that implies the relative risk of change taking place. In the context of this study, "risk" does not have a negative meaning but implies the possibility of change influenced mainly by the introduction of external APIs.

DV: Emergence of new clusters (t)	Model
Degree of external APIs (t-1)	0.048 (0.001)***
Frequency of external APIs (t-1)	0.006 (0.001)***
API age (t-1)	0.001(0.0001)***
Number of APIs provider offering (t-1)	- 0.013 (0.002)***
Constant	44.123(0.328)***
N	45,059
Number of group	344
Wald chi-square	3938.44

*: $p < 0.1$, **: $p < 0.05$, ***: $p < 0.01$

Table 2. Panel weibull survival model

In order to specifically understand the individual role of predictors in the change, we conducted a survival analysis. Table 2 shows the result. The coefficients indicate the probability of being API in the emergence of new clusters.

In our survival model, the degree of external APIs shows 4.8% increase at the 0.01 level. The positive sign of the result shows that external APIs with a high degree is more likely to become APIs in the newly emerged clusters in the following month. Thus, H1 is supported. The frequency of external API shows 0.6% increase at the 0.01 level. The positive sign represents that APIs with a high frequency is more likely to become APIs in the newly emerged clusters in the following month. Thus, H2 is supported. The result of H1 and H2 implies that the increased range of external APIs' connectivity is the key to diversify the structure. More specifically, by comparing the magnitude, external APIs that are widely connected with many others can affect the emergence of new clusters more than APIs that are frequently connected with small numbers. Thus, we can understand that APIs with high degree have more tendencies to lead the emergence of new clusters than within-cluster differentiation compared to APIs with high degree.

The result for age shows 0.1% increase at the 0.01 level. APIs stayed longer in a digital ecosystem will have more probability to interact with diverse APIs. Thus, H3 is supported. This implies that new external APIs are limited to generate a group of new add-on products. They need some time to be exposed to third party developers for the use in add-on products.

External APIs offered from companies that produce a large number of APIs are not necessarily likely to play a critical role in the emergence of new clusters in the model. The result shows a negative coefficient at the 0.01 level. Thus, H4 is not supported in the model. However, the result implies that individual API's functional usefulness is more important than the reputation and size of the platform company that offers the large number of APIs. The number of APIs that companies produce a large number of APIs represents opportunities to be relatively exposed and recognized a lot by third-party developers. Thus, we can consider that the bigger the company, the greater possibility of API use in add-on products. However, the analysis shows that APIs offered by small size company have an enough opportunity to increase the number of function calls of its API if the function of API is unique and useful for the creation of new add-on products.

Discussion

We explore the influence of newly introduced external APIs on the structural mutation of add-on product cluster in a digital ecosystem. As new APIs continuously join and actively contribute to the growth of a digital ecosystem, the platform system shows a dynamic and generative network structure that transforms from a single hierarchical structure to unanticipated multi-layered hierarchies. To explore the role of external APIs in this structural change, this paper adopts an evolutionary network perspective. The results offer a complementary view to previous studies on digital innovation. The generative nature of digital technologies creates change in such a way as not to limit the use of existing technologies but to increase use across different domains by continuously constructing a unique growth pattern in a focal platform system. This paper goes beyond the existing methodological lens to explore the dynamic change of a digital ecosystem. The various applied network analyses help to predict the structural changes related to the connection patterns among digital APIs. This method allows us to think of how much the landscape of a digital ecosystem can be changed, as the number of new external digital APIs and connection patterns are changed and to specify how a few critical external APIs affect the extent and pattern of landscape change. This paper offers guidance for future researchers considering an evolutionary network study to explore how network property and each node's attributes can be used to predict change. This method can ultimately help expand current digital ecosystem studies in order to understand the influence of various attributes on structural change.

This study offers the first empirical support to identify how changes in the landscape of a digital ecosystem can be systematically studied and how external technologies can lead to structural change by becoming a part of existing components. In particular, unlike the previous studies on ecosystems, we assume that the change of landscape mainly comes from the infinite number of heterogeneous components which create multiple hierarchical structures. The findings suggest that without introduction of external resources within the boundary of the platform owner's control, the ecosystem may not grow as dynamically as it does with them. External APIs play a critical role in creating functional diversity across multiple types of digital products in the growth of a digital ecosystem. Specifically, we found that broad connection with others is the key for change. Furthermore, the size of API providers does not necessarily influence the actual usefulness of their products. These findings demonstrate that a new platform provider or API provider can gain market competitiveness against large platform providers by providing a functionally useful add-on product. Once the product's technological feature is acknowledged, new firms can make their own market.

The generative nature of digital technologies shows different patterns of innovation than we have previously experienced. In particular, a digital ecosystem changes the vertically integrated hierarchical system structure designed by a platform owner into a horizontally coordinated, multi-layered system structure generated by third-party developers and contributed to by multiple digital ecosystems. The introduction of new technologies into a focal platform system changes the expected incremental innovation pattern into dynamic combinable patterns that cannot be anticipated. This paper explores the dynamic mechanism by specifically investigating the role of newly introduced heterogeneous digital technologies in a system. The findings allow us to think about unique and unexplored evolutionary forces for innovating a platform system.

Even though this study offers unique theoretical and methodological contributions for many other disciplines, there are some limitations. This paper captures the degree changes of each component that lead to the evolution of network structure through investigating the dependency between network properties and the attributes of components. However, we focused on a limited number of attributes to represent the nature of each node due to the nature of a single digital ecosystem.

This paper offers a new insight to understand the evolutionary changes in a digital ecosystem using the logic of the generative feature of digital technology and describes the main evolutionary forces and basic mechanism of structural change in a systematic way. In addition, the cutting-edge computational technique used in this study supports the empirical evidence of the theoretical argument. Therefore, this study provides a new way to explore the role of external digital technologies in discontinuous innovation patterns in a digital ecosystem.

Conclusion

A digital ecosystem represents the most dynamic aspects in the study of innovation. This dynamism derives from unbounded systemic features that lead to unpredictable evolutionary patterns. However, within the current innovation discourse it is difficult to fully explain the phenomenon of a digital ecosystem, as current studies of innovation are based on the assumption that systems are bounded and controlled by a central authority. This study offers a new perspective to examine these evolutionary dynamics and the specific mechanisms that produce them. Specifically, the unique findings of this study such as the impact of external APIs on a focal platform system offer a new perspective. However, this study is limited because it explored a single digital ecosystem. By focusing on more than two ecosystems in the future study, we can understand the dynamic growth of digital platform systems in a more specific way. This is an initial step in exploring the dynamic nature of various aspects of digital ecosystems in future studies.

References

Agarwal, R., Echambadi, R., Franco, A. M., and Sarkar, M. 2004. "Knowledge Transfer through Inheritance: Spin-out Generation, Development, and Survival," *The Academy of Management Journal*, pp. 501-522.

Arthur, W. B. 2009. *The Nature of Technology: What It Is and How It Evolves*. Free Press.

Baldwin, C. Y., and Clark, K. B. 2000. "Design Rules, Vol. 1: The Power of Modularity".

Barabási, A. L., and Albert, R. 1999. "Emergence of Scaling in Random Networks," *Science* (286:5439), pp. 509-512.

Boland, R. J., Lyytinen, K., and Yoo, Y. 2007. "Wakes of Innovation in Project Networks: The Case of Digital 3-D Representations in Architecture, Engineering, and Construction," *Organization Science* (18:4), pp. 631-647.

Borgatti, S. P., and Everett, M. G. 2000. "Models of Core/Periphery Structures," *Social Networks* (21:4), pp. 375-395.

Bornholdt, S., and Sneppen, K. 1998. "Neutral Mutations and Punctuated Equilibrium in Evolving Genetic Networks," *Physical Review Letters* (81:1), pp. 236-239.

Csermely, P., London, A., Wu, L.-Y., and Uzzi, B. 2013. "Structure and Dynamics of Core/Periphery Networks," *Journal of Complex Networks* (1:2), pp. 93-123.

Cusumano, M. A., and Gawer, A. 2002. "The Elements of Platform Leadership," *MIT Sloan Management Review* (43:3), pp. 51-58.

D'Souza, R. M., and Nagler, J. 2015. "Anomalous Critical and Supercritical Phenomena in Explosive Percolation," *Nature Physics* (11:7), pp. 531-538.

Doncheva, N. T., Assenov, Y., Domingues, F. S., and Albrecht, M. 2012. "Topological Analysis and Interactive Visualization of Biological Networks and Protein Structures," *Nature Protocols* (7:4), pp. 670-685.

Draghi, J. A., Parsons, T. L., Wagner, G. P., and Plotkin, J. B. 2010. "Mutational Robustness Can Facilitate Adaptation," *Nature* (463:7279), pp. 353-355.

Eaton, B., Elaluf-Calderwood, S., Sørensen, C., and Yoo, Y. 2015. "Distributed Tuning of Boundary Resources: The Case of Apple's Ios Service System," *MIS Quarterly* (39:1), pp. 217-243.

Eisenmann, T., Parker, G., and Van Alstyne, M. 2011. "Platform Envelopment," *Strategic Management Journal* (32:12), pp. 1270-1285.

Eisenmann, T., Parker, G., and Van Alstyne, M. W. 2006. "Strategies for Two-Sided Markets," *Harvard Business Review* (84:10), p. 92.

Gawer, A., and Cusumano, M. A. 2014. "Industry Platforms and Ecosystem Innovation," *Journal of Product Innovation Management* (31:3), pp. 417-433.

Ghazawneh, A., and Henfridsson, O. 2012. "Balancing Platform Control and External Contribution in Third-Party Development: The Boundary Resources Model," *Information Systems Journal* (22:2), pp. 173-192.

Horvath, S. 2011. *Weighted Network Analysis: Applications in Genomics and Systems Biology*. Springer Verlag.

Horvath, S., and Dong, J. 2008. "Geometric Interpretation of Gene Coexpression Network Analysis," *PLoS Computational Biology* (4:8), p. e1000117.

Johnson, S. L., Faraj, S., and Kudaravalli, S. 2014. "Emergence of Power Laws in Online Communities: The Role of Social Mechanisms and Preferential Attachment," *MIS Quarterly* (38:3), pp. 795-808.

Lacy, R. C. 1987. "Loss of Genetic Diversity from Managed Populations: Interacting Effects of Drift, Mutation, Immigration, Selection, and Population Subdivision," *Conservation Biology* (1:2), pp. 143-158.

Langfelder, P., Zhang, B., and Horvath, S. 2008. "Defining Clusters from a Hierarchical Cluster Tree: The Dynamic Tree Cut Package for R," *Bioinformatics* (24:5), pp. 719-720.

Lynch, M. 2002. "Intron Evolution as a Population-Genetic Process," *Proceedings of the National Academy of Sciences* (99:9), pp. 6118-6123.

Lynch, M. 2010. "Evolution of the Mutation Rate," *Trends in Genetics* (26:8), pp. 345-352.

Ohta, T., and Kimura, M. 1969. "Linkage Disequilibrium at Steady State Determined by Random Genetic Drift and Recurrent Mutation," *Genetics* (63:1), p. 229.

Parker, G. G., and Van Alstyne, M. W. 2005. "Two-Sided Network Effects: A Theory of Information Product Design," *Management Science* (51:10), pp. 1494-1504.

Radicchi, F. 2015. "Percolation in Real Interdependent Networks," *Nature Physics* (11:7), pp. 597-602.

Ravasz, E., and Barabási, A. L. 2003. "Hierarchical Organization in Complex Networks," *Physical Review E* (67:2), p. 026112.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A. L. 2002. "Hierarchical Organization of Modularity in Metabolic Networks," *Science* (297:5586), pp. 1551-1555.

Rombach, M. P., Porter, M. A., Fowler, J. H., and Mucha, P. J. 2014. "Core-Periphery Structure in Networks," *SIAM Journal on Applied Mathematics* (74:1), pp. 167-190.

Sanchez, R., and Mahoney, J. T. 1996. "Modularity, Flexibility, and Knowledge Management in Product and Organization Design," *Strategic Management Journal*, pp. 63-76.

Sosa, M. E., Eppinger, S. D., and Rowles, C. M. 2004. "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development," *Management Science*, pp. 1674-1689.

Spears, W. M. 1992. "Crossover or Mutation?," *FOGA*, pp. 221-237.

Tushman, M. L., and Anderson, P. 1986. "Technological Discontinuities and Organizational Environments," *Administrative Science Quarterly*, pp. 439-465.

Tushman, M. L., and Rosenkopf, L. 1992. "Organizational Determinants of Technological-Change-toward a Sociology of Technological Evolution," *Research in Organizational Behavior* (14), pp. 311-347.

Ulrich, K. 1995. "The Role of Product Architecture in the Manufacturing Firm," *Research Policy* (24:3), pp. 419-440.

Um, S., Yoo, Y., Wattal, S., Kulathinal, R., and Zhang, B. 2013. "The Architecture of Generativity in a Digital Ecosystem: A Network Biology Perspective,").

Wagner, G. P. 2003. "Evolutionary Genetics: The Nature of Hidden Genetic Variation Unveiled," *Current Biology* (13:24), pp. R958-R960.

Wagner, G. P., and Altenberg, L. 1996. "Perspective: Complex Adaptations and the Evolution of Evolvability," *Evolution*), pp. 967-976.

Wagner, G. P., Pavlicev, M., and Cheverud, J. M. 2007. "The Road to Modularity," *Nature Reviews Genetics* (8:12), pp. 921-931.

Watts, D. J., and Strogatz, S. H. 1998. "Collective Dynamics of 'Small-World'networks," *Nature* (393:6684), pp. 440-442.

Woodard, C. J. 2008. "Platform Competition in Digital Systems: Architectural Control and Value Migration".

Xia, Y., and Levitt, M. 2002. "Roles of Mutation and Recombination in the Evolution of Protein Thermodynamics," *Proceedings of the National Academy of Sciences* (99:16), pp. 10382-10387.

Yip, A. M., and Horvath, S. 2007. "Gene Network Interconnectedness and the Generalized Topological Overlap Measure," *BMC Bioinformatics* (8:1), p. 22.

Yoo, Y. 2013. "The Tables Have Turned: How Can the Information Systems Field Contribute to Technology and Innovation Management Research?," *Journal of the Association for Information Systems* (14:5), p. 227.

Yoo, Y., Boland, R. J., Lyytinen, K., and Majchrzak, A. 2012. "Organizing for Innovation in the Digitized World," *Organization Science* (23:5), pp. 1398-1408.

Yoo, Y., Henfridsson, O., and Lyytinen, K. 2010. "Research Commentary---the New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Information Systems Research* (21:4), pp. 724-735.

Zhao, W., Langfelder, P., Fuller, T., Dong, J., Li, A., and Hovarth, S. 2010. "Weighted Gene Coexpression Network Analysis: State of the Art," *Journal of Biopharmaceutical Statistics* (20:2), pp. 281-300.