The Evolution of Digital Ecosystems: A Case of WordPress from 2004 to 2014[1]

SungYong Um
School of Computing
National University of Singapore
sungyong@nus.edu.sg

Youngjin Yoo
Weatherhead School of Management
Case Western Reserve University
yxy23@case.edu

Sunil Wattal
Fox School of Business
Temple University
swattal@temple.edu

Abstract

Digital ecosystems are dynamic: they grow and evolve as new firms join the ecosystem. Yet, the way they evolve over time is not clearly understood. We draw on an evolutionary network approach to explore the evolutionary pattern of a digital ecosystem. In particular, we discover that the changing combination of existing digital components, interaction of which forms a complex bipartite network, drive the changes in the topological structure of a digital ecosystem over time. To formally test our ideas, we hypothesize the impact of network properties on the evolution of a digital ecosystem, and test them using a data set collected from WordPress.org. We used text mining on the source code data of WordPress plug-ins created from 2004 to 2014 and extracted the list of all API (Application Programming interface) used in these plug-ins. We then explore how the changes in the pattern of combinations of APIs drive the generativity of the platform as new plug-ins continue to emerge in the ecosystem over time. Our findings suggest that the evolution of a digital ecosystem represents a distinct structural interaction derived from the generative nature of APIs. A structural analysis shows that the rate of innovation does not necessarily increase though the number of APIs in a digital ecosystem increases, as the role of third party developers is limited to diversity the heterogeneity of a digital ecosystem by their own. Instead, platform providers need to consider the time point when they are going to open their core technologies and introduce heterogeneous technology of others.

Keywords: digital ecosystem, complex network evolution, generativity

---

[1] Funding for this research provided in part by National Science Foundation (SES – 1261977 and VOSS-1120966) and CIFREG. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funders.

Introduction

We are witnessing the rapid growth of software-based digital open systems that are highly generative (Boudreau 2012; Yoo et al. 2012). Companies like Apple, Google and Facebook compete not only based on the features of their products, but also on the size and the heterogeneity of their respective ecosystems. Digital ecosystems are dynamic as new firms continue to join the ecosystem. An ecosystem's ability to attract new firms and continue to evolve is of strategic importance for those who create and own the ecosystem as well as those who decide which ecosystems to join (Ceccagnoli et al. 2012; Eaton et al. 2015). With increasing digitization, digital ecosystems are moving beyond traditional software products such as web services or mobile services and integrating into physical products such as automobiles, watches, and televisions. Therefore, understanding how such ecosystems evolve over time and what drives innovations of and in such ecosystems is becoming critically important for most firms.

Although these digital ecosystems are based on modularity (Baldwin and Clark 2000; Langlois 2002; Schilling 2002) and thus the way they evolve is in part influenced by the logic of modularity (Tiwana 2015), they differ from traditional physical products that follow modular architecture in some important ways. Even though some firms have opened up their product design architectures to enable innovations by third-party developers (Chesbrough 2003), the product is designed first by the focal firm, which controls the design rules (Baldwin and Clark 2000; Henderson and Clark 1990; Simon 1962). In this scenario, therefore, the role of third-party developers is limited to incremental innovations within a design hierarchy through mix and match (Baldwin and Von Hippel 2010). This produces a typical punctuated pattern of product innovations where a radical change of a fundamental product architecture change is followed by a long stretch of incremental changes in modules (Anderson and Tushman 1990; Henderson and Clark 1990).

Unlike traditional modular architecture, software-based ecosystems are based on layered modular architecture in which the architecture is not given by a focal firm, but rather emerges through on-going uncoordinated actions by heterogeneous third-party developers (Yoo et al. 2010). A software-based digital ecosystem refers to the combination of a *digital platform* -- an extensible software code base that provides a limited number of *digital components* such as Application Programming Interface (API) -- and a collection of heterogeneous *add-on digital products* built by third-party developers using provided digital components (Tiwana et al. 2010). Yoo et al. (2010) argue that layered modular architecture enables digital products to be highly generative. At the same time, it can potentially lead to an increase in complexity in the pattern of digital evolution as third-party developers do not have a fixed set of design rules to follow (Yoo et al. 2010). However, despite the lack of central control and a fixed set of design rules, innovation in a digital ecosystem shows a remarkably ordered underlying structural pattern (Um et al. 2013). To further our understanding of this issue, we explore how this structural pattern emerges and evolves over time (Tiwana et al. 2010).

In the context of a digital ecosystem, add-on digital products ("digital products" hereafter) are generated from the combination of digital components, which are combined together to deliver a set of coherent functions (Yoo et al. 2012). Seemingly unrelated digital products form a network over time based on the digital components that they share. Furthermore, we can discover clusters of digital products based on how these digital products share different digital components. Thus, such a network can be seen as an underlying architecture that gives birth to the generativity we observe in a digital ecosystem. An evolutionary network perspective provides a systematic theoretical lens to understand how structural patterns in a network evolves over time (Ravasz et al. 2002; Stuart et al. 2003). Specifically, we explore how digital products in a digital ecosystem form a dynamic bipartite network by looking at how they share common digital components and how that sharing pattern changes over time (Dhar et al. 2014). From this network, we create a topological overlap to identify clusters of digital products that share similar sets of digital components. Given the open nature of a digital ecosystem, new digital components continue to emerge and the existing digital components mutate over time. Such changes in the digital components then cause changes in the way they are combined to produce digital products, causing the changes in the clusters of these products. As a result, some clusters split, while others expand or mutate. Thus, the way clusters of digital products in a digital ecosystem form and evolve through changes in the underlying digital components represents the fundamental architecture of the generativity of a digital ecosystem. Therefore, we explore how the changes in digital components affect the evolution of the structure of clusters of digital products in a digital ecosystem.

The evolutionary logic of digital innovation is still under-developed, as previous studies have mainly focused on the static nature of digital innovation (Boland et al. 2007; Yoo et al. 2012; Yoo et al. 2010). This paper focuses on the evolution of digital products in a digital ecosystem to understand the evolutionary logic of digital innovation. Specifically, this paper asks:

1) What is the evolutionary pattern of digital innovation in a digital software-based ecosystem?

2) How does the combination of digital components in a digital ecosystem affect the evolutionary pattern of digital innovation over time?

We answer these questions with a data set collected from the source code of plug-ins from WordPress.org ("WordPress" from hereafter). WordPress offers the world's largest blogging service. It is structured as a digital ecosystem where third party developers can combine APIs offered from WordPress and other web service providers (such as Google or Facebook) to create plug-ins. In this empirical context, an ecosystem consists of a digital platform (WordPress), digital components (APIs) and digital products (plug-ins). These plug-ins form an ever-changing landscape of clusters through changing combinatorial patterns of APIs. The remainder of the paper proceeds as follows: we first review existing literature on the evolution of combinatorial innovation and generativity. Second, we introduce a biological evolutionary network approach to explore the evolution of generativity. Third, we describe the empirical model and highlight its main results. Finally, we discuss the theoretical and methodological implication of this study.

Literature Review

Digital innovation in a layered modular architecture

Innovation evolves from the combination of existing components that give a product new features or uses (Fleming and Sorenson 2004; Nelson and Winter 1982). A platform is modularized into decomposable components for innovation (Simon 1962). A platform owner can create a digital ecosystem to achieve innovation beyond the owner's ability by opening the interface of a product to third-party developers. The increased size of a digital ecosystem increases the possible number of combinations for innovation

(Fleming 2001; Kauffman 1993). A platform owner expects to improve the functional flexibility of the platform through the larger number of digital products developed by third-party developers (Sanchez and Mahoney 1996), but also wants to control product design and its evolution through the architectural control (Baldwin and Woodard 2009).

Innovations in a software-based digital ecosystem represent a unique design pattern that challenges the evolutionary pattern of innovation in a modular architecture. With a layered modular architecture, a digital platform does not have a given fixed design boundary for the design of digital products so that any digital components in a digital ecosystem can be combined in a new way (Yoo et al. 2010). Therefore, the combination of digital components in the layered modular architecture represents a complex and generative innovation pattern, as innovation occurs in different layers at the same time in often unexpected ways (Adomavicius et al. 2008; Benkler 2006).

Innovations in a layered modular architecture take place through the generation of digital products by third-party developers who participate in a digital ecosystem (Boudreau 2012; Ceccagnoli et al. 2012) . A digital product or an add-on to a layered modular architecture is not designed based on a fixed standardized interface. Instead, digital components from different design hierarchies can be used in building digital products, as they can co-exist in a platform due to the product's agonistic nature (Yoo et al. 2010). This means that a digital product features a combined set of digital components whose interaction is differently defined by third-party developers with respect to an intended coherent function. Thus, the recombinatorial nature of digital innovation in a digital ecosystem forms the genetic foundation of generativity defined as the reproductive capacity of digital components for unprompted and uncoordinated changes by a large, varied, and uncoordinated audience (Benkler 2006; Fleming et al. 2007b; Zittrain 2006).

An evolutionary network perspective on digital innovation

An evolutionary perspective has been a useful theoretical framework in exploring the evolution of technology (Fleming and Sorenson 2001; Levinthal 1997; Tiwana 2015). The basic notion is based on the holistic view that any technological innovation takes place with respect to existing design components

(Fleming 2001). Innovation as the combination of new or existing design components has been compared to the adaptive process of genetic recombination in an organism (Kauffman 1993). In particular, the number of design components and their interactions have been mainly used as the critical underlying properties to explain how combinatorial innovations take place. The number of available design components represents the diversity in an ecosystem. The interaction among design components indicates how each design component is inter-connected in the combination.

However, the existing evolutionary theoretical lens on innovation (Fleming and Sorenson 2001; Kauffman 1993; Levinthal 1997) does not consider the unique feature of digital innovation because the boundary of a digital ecosystem is often unspecified a priori (Boudreau and Jeppesen 2014). Unlike a traditional physical system whose system boundary is pre-specified, digital ecosystems allow new components to shape and re-shape the existing boundary of the ecosystems as they split, merge and mutate over time. Particularly, a combined set of components used across different cluster boundaries needs to be considered in order to reflect the generative feature of digital components (Goldberg et al. 1993; Simon 1962).

To fill this gap, this paper adopts an evolutionary network perspective (Dhar et al. 2014; Ravasz and Barabási 2003). In particular, this paper focuses on the evolution of topological structure in a network to complement the understanding of a system from a *decomposable system* based on architectural knowledge (Baldwin and Clark 2000) to a *developmental combinable system* (Holland 1975; Wagner and Altenberg 1996) derived from the generative nature of system-agnostic digital components (DeLanda 2013; Yoo et al. 2010).

An evolutionary network perspective explains the topological change of clusters in a network, as a network represents the hierarchical order of sub-divided networks depending on the interaction of nodes (Strogatz 2001). A node represents an object such as an organism or a cell including various components. For example, an organism with a single cell shows structural and functional heterogeneity that can be separated into a set of networked genes which work together so that the basic gene components produce different cell types over time (Wagner et al. 2007). The variation of organisms resulting from genetic variations represents hierarchical sub-function boundaries, or clusters, based on a topological overlap structure, which is constructed by the interactions of commonly shared genes (Wagner and Altenberg

1996). Thus, highly topologically overlapped objects are positioned in the center of nested hierarchies in a cluster, and other objects share the same components combined with other components in a cluster.

Applied in the context of digital innovation, the topological overlap generates a nested-hierarchy among digital products developed by various digital components in a digital ecosystem. Seemingly unrelated digital products are networked together based on the common usage of digital components and form *clusters*. As such, digital products display a topologically overlapped structure with neighbor digital products in a network (Ravasz et al. 2002). Highly topologically overlapped digital products form a hierarchical order with other digital products based on a set of commonly shared digital components in a cluster. The topological structure in the study of evolutionary network emphasizes the pattern of unexpected combinatorial change in response to the changes of existing components and the emergence of new components (Wagner and Altenberg 1996). Therefore, the evolution of digital innovation can be understood based on the change of a network's nested hierarchical structure (or the evolution of clusters) over time depending on the combinatorial pattern of digital components over time.

Theory Development

The basic mechanism of evolutionary pattern

Non-linear interaction breaks the inherent combinatorial patterns taking place among given digital components. Each digital component has a unique functional feature distinct from others. In addition, the generative nature of digital components emphasizes the equal importance of each digital component. However, digital components are not equally used in digital products in non-linear interaction (Tushman and Rosenkopf 1992). Some components are repetitively used for the generation of new products, while others are used infrequently.

A digital ecosystem consists of both internal digital components that are provided by a focal platform owner and external digital components that are not provided by the owner. Furthermore, in a digital ecosystem, a certain set of individual digital components is frequently used together, forming a tightly coupled building block for the entire ecosystem or for a specific cluster. To specifically understand the dynamic combinatorial pattern, we categorizes digital components based on the degree of usage (Um et al. 2013). Based on this, the entire array of digital components in a digital ecosystem can be divided into universal core components, sub-cluster core components, and periphery components depending on the usage (Borgatti and Everett 2000; Csermely et al. 2013; Rombach et al. 2014)

Some of these tightly coupled building blocks are entirely made of internal digital components and used most frequently throughout the ecosystem. Thus, we refer to these as *universal core digital components*. Universal core components define the basic building blocks of any digital products in a platform. Universal core components are highly connected with other components throughout the entire ecosystem. Other building blocks of digital components that are often combinations of internal and external components are not used as frequently and universally as universal cores in a platform, but independently form their own structure as a foundation of individual clusters. Thus, we refer to them as *sub-cluster core digital components*. Sub-cluster core components are highly connected with other components within each cluster, creating the heterogeneity among clusters. The rest of the digital components are often used to modify the universal core components and sub-cluster core components in order to diversify the function of digital products within a cluster. These are referred to as *periphery digital components*. Periphery digital components produce variety within a cluster. Periphery components are loosely connected components mainly with sub-cluster cores and create structural and functional variability within each cluster

Characterized in this way, there are two different options for the evolution in a digital ecosystem. If the combinatorial pattern of digital components from the selection process does not change over time, the evolutionary trajectory of the clusters will remain the same under the existing hierarchy. However, if the combinatorial pattern changes, the network has a change of topological structure in a network with different evolutionary trajectories. With this view of the combinatorial interaction of digital components and the topological structure of clusters, this paper takes an evolutionary network perspective to focus on the breadth and depth of a network as a way to explore how the rate of evolution of digital innovation changes in a digital ecosystem (Barabási and Oltvai 2004; Ravasz et al. 2002; Wagner et al. 2007). In this paper, the breadth of a network implies the size of a network that shows the increased number of digital products, while the depth of a network represent the heterogeneity of a network that is explained by the degree of nested hierarchy among digital products understood by the topological overlap measure (Horvath 2011; Langfelder et al. 2008).

To understand the structural evolution of a digital ecosystem, this study focuses on the changes in clusters formed by groups of digital products in a digital ecosystem. A node is a digital product whose function is expressed from the combined usage of digital components. Nodes are connected and clustered with one another owing to the shared digital components in a digital product network. Figure 1 offers a simple illustration of the evolutionary pattern of a digital ecosystem.

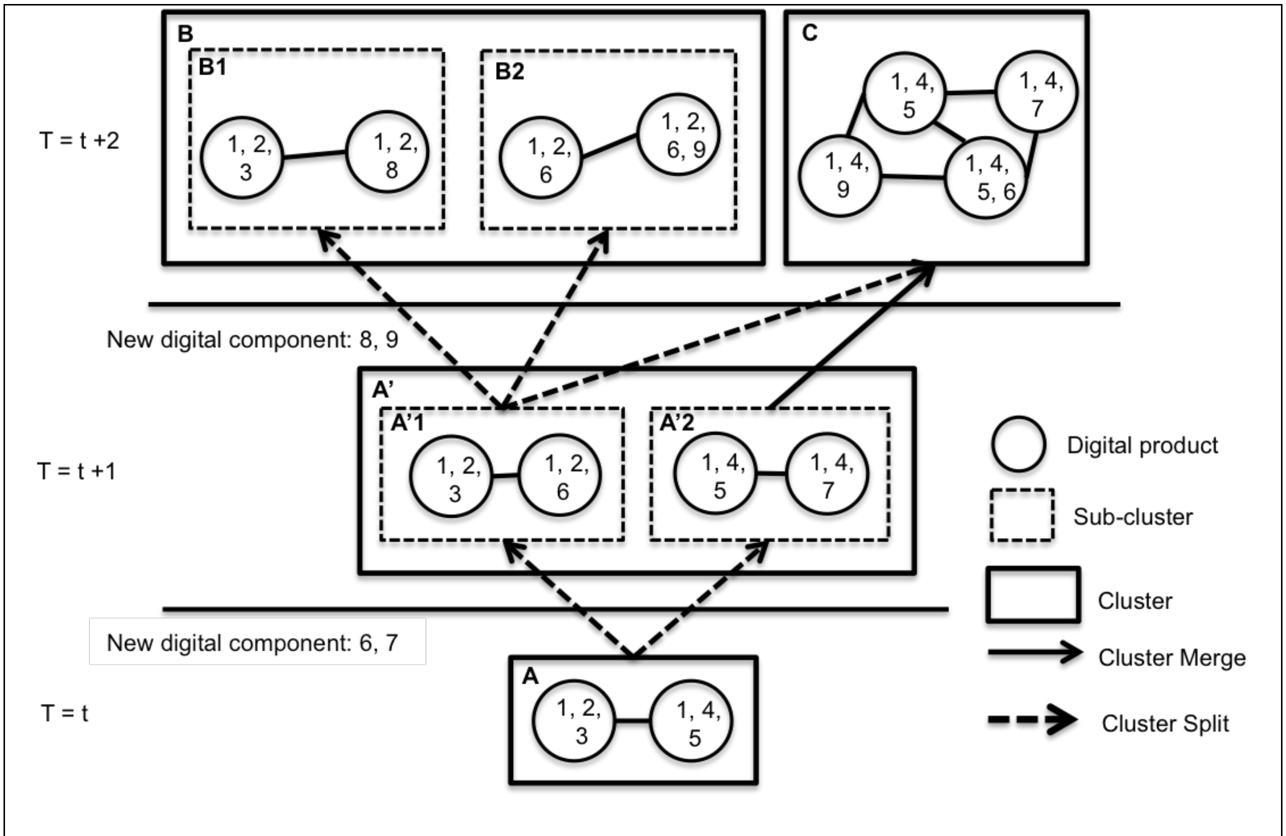


Figure 1. The Basic Mechanism of Evolutionary Pattern

Five different digital components (represented by the numbers 1 to 5) exist at the initial stage of a network at time t. Two different digital products (shown as circles) are developed from the two different combinations of the five digital components. The box of Figure 1 at the bottom in Figure 1 represents a cluster that is grouped from the connection of a commonly shared digital component (digital component 1) in the initial time period. The cluster (A') including the two sub-clusters in the middle (at time t+1) represents the second generation of cluster as a result of evolution that takes place by the introduction of new digital components (digital component 6 and 7). Two new digital products that are generated from the combination of new digital components form the two sub-clusters A'1 and A'2 that are topologically categorized depending on the same usage of digital components: digital component 1 and 2 for A'1 and digital component 1 and 4 for A'2. The topological overlap from the commonly shared digital component (digital component 1) enables the two sub-clusters to be grouped in the same cluster A' at time t+1.

The two clusters at the top represent the third generation of clusters as a result of another evolution process derived from 1) the introduction of new digital components (digital component 8 and 9) which are acquired from a digital ecosystem and 2) the existing digital components that are recombined with new digital products to create more new digital products. The cumulative number of digital products in the sub-cluster A'1 and A'2 makes the cluster A' evolve with different topological overlaps. Cluster A' is split: one part of the split forms a new cluster B based on the digital components 1 and 2. In particular, cluster A'1 is split into B1 and B2 where B1 forms a sub-cluster depending on the shared digital components 1 and 2, while B2 makes a sub-cluster that constructs the topological overlap from digital components 1, 2, and 6. Thus, sub-clusters B1 and B2 are topologically linked with each other based on digital component 1 and 2 that forms the nested hierarchy of cluster B. Another part of the split takes place in cluster C. Here, cluster C is generated from digital products from sub-cluster A'2 and the combination of digital products with new digital component 9 and existing digital component 6 that migrated from sub-cluster A'1. All digital products are strongly connected with digital components 1 and 4 that form a distinct topological overlap pattern compared to digital products in cluster B.

The hypothetical example in Figure 1 gives an insight into how the evolutionary pattern can be understood from the topological recombination derived from new digital products in clusters in a digital product network. In particular, from the example, we can learn the characteristic of digital components in a network. At time t+2, digital component 1 is used in two clusters, B and C. As such, digital component 1 is considered a universal core component. In cluster B, digital component 2 is used in two different sub-clusters, B1 and B2, while digital component 4 is commonly used in all digital products in cluster C. Thus, digital components 2 and 4 are treated as sub-cluster core components. The rest of the digital components used in each cluster are understood as periphery digital components. Based on this conceptual notion, this paper focuses on the evolutionary pattern across clusters and the combinatorial pattern of digital components in clusters in a network in order to explain the mechanism of how the evolution of digital innovation takes place.

From Figure 1, we focus on a reciprocal set of three interactions that have mutual effects changes one to the other described in Figure 1: 1) digital components, 2) the size of cluster, and 3) the size of sub-cluster. First, the number of digital components changes as new ones are continuously introduced in a digital ecosystem. As such, the size of digital components increases over time. In addition, the functional heterogeneity of digital components increases, as various functions emerge over time. The degree of digital component usages determines the different structural layers of digital components. The generative nature of digital components can be categorized into a different layer over time. Some of popular digital components can be moved from periphery to sub-cluster section. Thus, the ratio of external digital components in the sub-cluster section is dynamic in nature.

Second, the size of cluster indicates the total number of digital products in each cluster. A cluster groups digital products in terms of the common usage of digital components in a network. As such, a cluster represents the distinct common combinatorial pattern among digital products that essentially shows the functional heterogeneity of digital products. However, the size of cluster in this paper simply represents the total number that the sum of digital products across all clusters is the same of the total number of digital products in a digital ecosystem. We expect to control the functional heterogeneity which come from the usage of digital component among digital products.

Third, the heterogeneity of sub-clusters represents the degree of functional heterogeneity. A cluster includes multiple different patterns in terms of digital component usages, and digital products can be grouped into various clusters. Thus, a sub-cluster is a nested cluster in a cluster grouped by the combination pattern of digital components. Sub-clusters show the nested hierarchy, as the commonly used digital components form the boundary of topological overlap that leads to hierarchical order among sub-clusters. Specifically, a few popular digital components have unique functional heterogeneity compared to other digital components that mostly combine with the popular ones. Thus, digital products typically include a few popular digital components positioned in the upper layer in a cluster, and the other digital products using various other digital components will be topologically located in the lower layer in the nested hierarchical structure. Therefore, the degree of nested hierarchy implies the degree of functionally heterogeneous digital component usages in digital products.

Hypotheses

We set up a null hypothesis that a digital ecosystem evolves without a specifically designed structural pattern over time from continuous non-linear interactions among existing digital components and newly emerged digital components. The connection patterns of digital products based on the shared usage of digital components form clusters that construct the structural pattern. These dynamics of continuously generated new structural patterns are distinct from the patterns in the past time period, as the size and heterogeneity of digital components in a digital ecosystem change over time. The non-linear interaction among digital components mainly takes place within the boundary of clusters in a network (Watts and Strogatz 1998). Some digital components can be universally adopted across different clusters, while others are mainly used within a cluster. The shared digital components in each cluster construct a hierarchical structure based on topological overlap (Ravasz et al. 2002). This overlap can be characterized as a nested hierarchy in which each cluster is derived from complex non-linear interactions depending on the type of digital components. As such, digital products that are composed of highly overlapped groups of digital components are positioned in the topologically upper layer in a digital product network structure, while the less overlapped ones are located in the lower hierarchical layer.

In order to find the impact of combinatorial patterns on the evolutionary trajectory, we focus on the interaction dynamics specifically between the internal and external sub-cluster core components and the periphery components. In particular, we do not consider the role of universal core components, as universal core components represent high density in the whole network that does not significantly change the nested hierarchical structure and diversity of digital components (Rombach et al. 2014). Furthermore, the number of universal core components is stable over time for all clusters and the topological structure of a cluster is constructed and changed mainly based on sub-cluster core components, as they are highly used in clusters in a network (Ravasz et al. 2002). We focus on the topological overlap structure of clusters (or nested hierarchy) in a network as a way to capture the change of the size and heterogeneity of a digital ecosystem. This paper explores the role of diverse digital components categorized into sub-cluster core and periphery on the structural change of a digital ecosystem. We do not significantly consider the functional role of each digital component. Instead, we consider the ratio of external digital components to sub-cluster sections in order to understand the impact of heterogeneous digital components on the structural change.

Sub-cluster core components are not as densely connected as universal core components. Unlike the universal cores, sub-cluster core components are functionally heterogeneous and diverse, as their functionalities do not need to be tightly related with a focal platform system. They can be used in various ways, and any digital component can become a sub-cluster core component. As such, several different combinations of the sub-cluster core components contribute to the generation of a new cluster of digital products. Each sub-cluster core component in a digital product represents a different degree of topological overlap in each cluster. Thus, sub-cluster core digital components have multiple sub-groups with respect to the number of individual digital components. The degree of topological overlap changes continuously when digital components combine with each other. Therefore, the increased number of sub-cluster core components influences the change of topological structure in a cluster. Specifically, a cluster with a large number of sub-cluster core components is likely to attract more digital products to the cluster in the future. Similarly, a cluster with a large number of sub-cluster core components is more likely to be divided into a number of sub-clusters within the cluster in the future. Thus, we hypothesize:

H1a*: The number of sub-cluster core components in a cluster positively affects the size of the cluster.*

Though digital components represent the agnostic to specific digital product design, the generative feature of a digital component does not mean that it can be used for anything. Each sub-cluster core component can be used individually depending on its functional fitness with other components for a coherent function. As such, the influx of new digital components into a cluster does not ensure the continuous increase of combinatorial patterns. Therefore, each sub-cluster core component has a different degree of topological overlap. If the new components have a functional fit with existing components, they will represent functional applicability, and new topological structures will also be constructed. However, if they do not fit functionally, they will not change the topological structure, and the rate of change will not always increase. The following is posited:

***H1b**: The impact of sub-cluster core components on the cluster size follows a curvilinear (inverted u-shaped) form.*

There are numerous functional types of periphery components in a digital ecosystem. The continuous generation of new digital products is possible in part because periphery components can be combined with other digital components to produce functionally unique digital products. Even though each digital product has a unique function, digital products can have a functional similarity if they share a periphery digital component belonging to the same functional type of external digital component. For example, if digital products include the Google Map API, they will commonly have a map function. Periphery digital components can be combined with both core and other periphery digital components. Specifically, universal core and sub-cluster core components can be combined with any functional types of periphery components. A digital product can include several periphery components if they are functionally fit with one another for the functional coherence of a digital product. For example, a digital product can be developed with universal core and sub-cluster core digital components with the combination of the Yahoo! Map Image API and the Google Map API to match image files on a map. As such, various combinations of periphery components are possible. Thus, repeatedly used periphery components can change the topological structure in clusters in a network. Periphery components can increase the size of a cluster in a network based on the numerous possible combinations. At the same time, repeated usages in a cluster change the topological overlap and influence the density of topological structure in a cluster in a network. For this reason, periphery components will contribute to the change of evolutionary pattern over time. Thus, we hypothesize:

***H2a**: The number of periphery digital components positively affects the size of the cluster.*

There are a huge number of digital components in a digital ecosystem. For example, the website www.programmableweb.com listed more than 14,000 digital components registered as of December 2015. Each digital component listed on the website has its own unique function. However, the generation of a totally new functional type is limited although similar but slightly different functions are frequently generated. Thus, the creation of new digital products through recombination can be limited. For example, third-party developers can use the Google Maps service for their digital products. Other platform services such as Yahoo! and Amazon provide similar map API services to third-party developers. As such, even though the number of digital components increases, functional diversity does not necessarily increase at the same time. Furthermore, functionally similar periphery digital components will not be equally used because developers have different preferences for various brands (Newman et al. 2002). A few map APIs in a digital ecosystem will be highly used, while the others will have low usage. Thus, the different number of usages of each periphery digital components does not influence the change of the evolutionary trajectory in a particular pattern. Thus, we hypothesize:

***H2b**: The impact of periphery components on the cluster size follows a curvilinear (inverted U-shaped) form.*

The role of digital components is not fixed in the introduction stage of digital components. The continuous usage of digital components changes the categorized role of digital components. Some of popular external digital components are placed in different layers by the degree of usages, as the usage of external digital components instigates more usages because their functional usefulness in digital products ultimately leads to continuous combinations of the other digital components over time. Thus, the ratio of external digital components depends on the number of combined other digital components that can be limited to within a cluster. Some popular external digital components are highly connected with other digital components, so they show high topological centrality compared to other combined digital components in a cluster. Their high centrality implies that the external digital components have enough of their own functional uniqueness to have their own group in a cluster. Thus, an increased ratio of external sub-cluster digital components indicates increased functional heterogeneity in a cluster. Thus, we hypothesize:

***H3a**: The ratio of external sub-cluster digital components positively affects the heterogeneity of sub-clusters of a cluster.*

The popularity of external digital components in sub-clusters implies that they are densely connected with other components not only in the sub-cluster section but also in the periphery section. Each external digital component has its own nested hierarchy in terms of its connection with other digital components. At the same time, the highly connected digital components in a sub-cluster section are also connected with other external digital components in a sub-cluster section. As such, the external digital components having nested hierarchies represent their hierarchical orders depending on their degree of influence on connection with others. Only a few external digital components that have more functional universality have more influence in a sub-cluster section. Therefore, the external digital components in a sub-cluster section do not have the same impact on the generation of digital products. Only a few have more influence than others to build a larger size of sub-cluster section. Therefore, the functional heterogeneity of sub-clusters will be highly related with a few highly connected external digital components that highly affect the increased number of digital products in clusters. Thus, we hypothesize:

***H3b**: The impact of external sub-cluster digital components on the heterogeneity of sub-clusters is negatively moderated by the size of the cluster.*

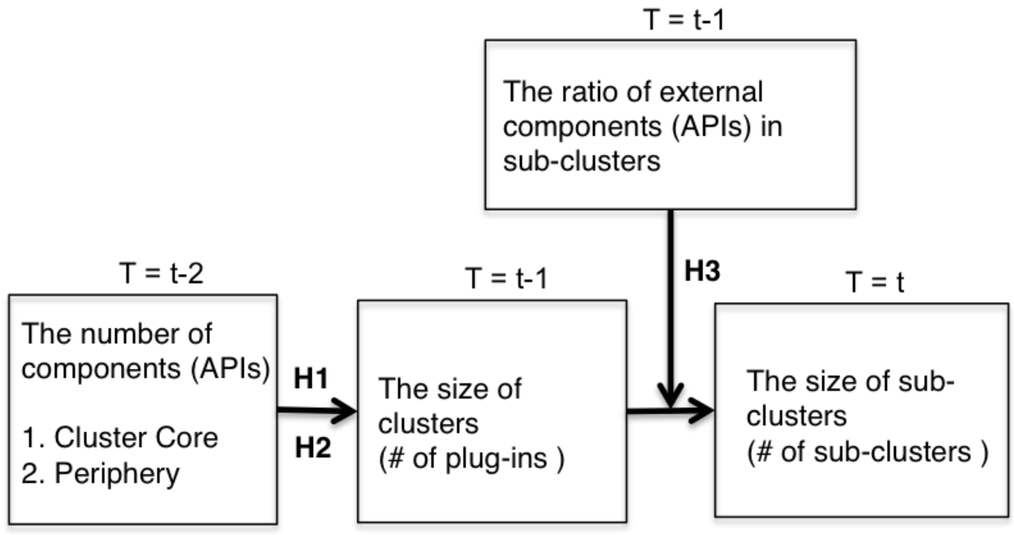


Figure 2. The model for the evolutionary Pattern

Analytical Approach

Co-expression network

We analyzed our data in a number of different steps. First, we capture the cluster dynamics in a network to specifically explore the evolution. Clusters are the result of the combination of digital components that construct the underlying structure of digital innovation. A cluster can be dependent on another cluster, as the same digital components can be used in different clusters. In particular, nodes of a network need to be fully connected to explore the dynamics of clusters (Watts and Strogatz 1998). We built a co-expression network of digital products to represent the expression of digital components (Stuart et al. 2003). A co-expression network is a weighted and undirected network (Zhang and Horvath 2005). It is effective in capturing the correlation derived from digital components used in each digital product. In particular, a co-expression network represents the typology of a network depending on the frequency of digital components. Hierarchical interactions from the topological overlap of the usage of digital components can be considered to explore network dynamics by using a co-expression network (Ravasz et al. 2002).

Clusters in a co-expression network are segmented based on the topological overlap measure of each digital product. Topological overlap indicates the degree of interconnectedness among digital products (Ravasz et al. 2002) in terms of the direct network of digital products and the number of shared digital components. It can be measured in the following way. The number of direct neighbors of a digital product i and j is defined by $k_i = \sum \quad a_{ij}$. k_i represents the correlation between a digital product and its neighbors depending on commonly used digital components. The number of digital components commonly used by digital products i and j is $\sum_{u \neq i,j} a_{iu} a_{ju}$. From the two, the topological overlap of each digital product is calculated as the following:

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

Statistical Model Specification

In this study, the basic unit of analysis is a specified cluster in a network where new digital products change the topological structure characterized by the pairwise connection of APIs. This study focuses on the change of connection pattern as new plug-ins are generated to capture the structural change (split and merge) of a network in terms of clusters. We can statistically identify the role of internal and external APIs in terms of the ratio of external APIs in the generation of plug-ins by extracting the information from the network analysis.

We use the two different types of variables to capture the evolution of digital innovation. First, we use the number of digital products in each cluster to measure the size of cluster. It directly captures the combinatorial dynamics of digital components in a digital ecosystem. Second, we used the number of nested hierarchies in a cluster to measure the size of sub-cluster. We can understand the change of topological structure in a cluster in terms of the functional heterogeneity of digital products in groups. We expect to explain the impact of combinatorial dynamics on the structural change in a cluster which represents an evolutionary pattern based on topological overlaps.

We use the number of sub-cluster core and periphery digital components as key independent variables in this study. The number of universal core digital components rarely changes, as it allows digital products to be adopted on the platform system. Thus, each model does not include a quadratic term for the universal core API. The square values of sub-cluster core and periphery components are used to explore the non-linear pattern of the evolution of combinatorial innovation in a digital ecosystem over time. We mainly use a panel OLS regression for model 1 to understand the evolutionary dynamics of digital products. Cluster and time fixed effects are used to control unobserved variables based on the Hausman test.

An Empirical Study

Data

We collected data from WordPress, focusing on plug-ins as an example to explore the evolutionary pattern of generativity in a digital ecosystem. The digital components on WordPress are APIs. One or more APIs are used to form a plug-in (digital product). We downloaded more than 100GB of source code data in text files from WordPress. To effectively capture API data, we developed a text-mining program written in Java that captured internal and external APIs used in all different versions of 23,895 plug-ins from January 2004 to December 2014. We constructed plug-in by API monthly matrixes to explore the interaction of APIs in plug-ins. The entries of each matrix are composed of binary values to represent which APIs were used for the expression of plug-ins. The two numeric values were followed by the NK landscape model (Kauffman 1993). '1' means the API is used in the plug-in, while '0' means it is not. All versions of the source code in each plug-in were analyzed. In January 2004, there were 86 plug-ins using 44 APIs. In December 2014, there were 23,985 plug-ins using 443 APIs, 113 of which are provided by WordPress, with the remaining 330 APIs offered by other platform providers such as Google, Yahoo, and Facebook.

We extracted available data about the age and the number of functional updates of each plug-in from the log files. The age of plug-ins was determined based on when the plug-ins were created as recorded in the source code history. Each plug-in is updated over time to resolve bugs and add new functions that are recorded in each log file. We extracted the information when the update took place using binary values. '1' indicates the time of update, while '0' shows that a plug-in is not updated. We also extracted developer data shown in each plug-in page in WordPress. The developer information tells us how many developers are involved in each plug-in. We also examined another digital ecosystem (www.programmableweb.com) to understand if the WordPress ecosystem represents a unique growth pattern. We collected the number of mash-up APIs that result when third-party developers freely create new APIs by combining heterogeneous functional types of APIs in the ProgrammableWeb ecosystem on a monthly basis from October 2005 to December 2014. These collected data can be used for an instrument variable that can control the independent variables such as plug-in age, the number of updates, and developer information.

Results

A Co-expression network

We used R to analyze a plug-in co-expression network to visually explore how the underlying structural pattern changed from 2004 to 2014 (Horvath 2011). We extracted the plug-in data for each cluster from 2004 to 2014 for further statistical analysis. The analyzed results of a network in three different years are provided in Figure 2.

First, each analysis includes a tree diagram above a network analysis in order to illustrate the hierarchical order of plug-ins with respect to the combinatorial pattern of APIs. The height of the tree diagram represents the degree of similar combinatorial patterns among plug-ins. The bottom of a tree indicates low similarity, while the top represents high similarity (e.g., sharing commonly used APIs). Plug-ins that include commonly used APIs are positioned on the top on each branch, while plug-ins with low interaction frequency APIs can be found at the bottom of a tree. A color bar below a tree diagram indicates the segment of clusters, which is calculated by a hierarchical clustering method (Langfelder et al. 2008). Each color in a color bar specifies clusters depending on combinatorial difference.

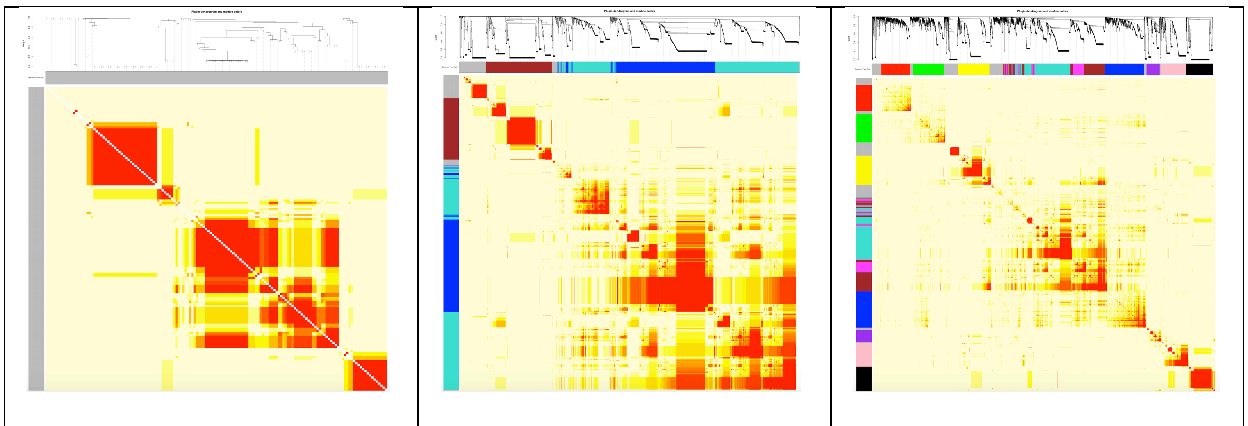
Second, a color-coded network below a color bar indicates a co-expression network analysis. Red color represents high similarity, while yellow shows low similarity. Thus, red color indicates densely connected regions, and yellow color represents sparsely connected areas. X-axis and Y-axis indicate plug-ins. We did not include the specific name of each plug-in, as more than 20,000 plug-ins were visually analyzed. A co-expression network analysis of different time periods represents how the structural pattern evolves over time through the change of the size and density of clusters. We can detect how many new plug-ins were generated through the change in the tree diagram.

Third, to understand how the structural change has taken place, we can check two different regions in each analysis. One is the diagonal region along a diagonal line, and the other is the off-diagonal region. A cluster that is specified by a color bar can include sub-clusters which represent a nested hierarchy. Several clusters on a diagonal line show 1) that there are certain structural patterns with respect to the combination of APIs, and 2) how plug-ins are connected within each cluster. Even within the boundary of a cluster specified by a color bar, some plug-ins are highly connected with one another. The different density of colors represents the degree of connectivity among plug-ins in each cluster. Connectedness in the off-diagonal region shows how plug-ins belonging to different clusters interact with each other across different clusters specified by a color bar, indicating interdependency across clusters.

Figure 2 illustrates the plug-in network in 2006, 2010 and 2014 to compare the size and structure of each network. Each plug-in is a node in the network. A plug-in co-expression network shows 0.39 cluster coefficient on average with 0.022 standard deviation from 2004 to 2014. The graphics show how the structure of a plug-in network evolves over time. First, from each tree diagram, we can understand the hierarchical relationship among plug-ins structured from the combinatorial pattern of APIs. In particular, seemingly unrelated plug-ins can be categorized depending on their usage of APIs, and some plug-ins became the parents of other plug-ins. We can detect that new branches are continuously generated over time, as new APIs in a digital ecosystem are used for new combinatorial patterns.

Second, as shown in the color bar below each tree diagram, the number of clusters increases from 1 to 11. Though the number of plug-ins is drastically increased, we can understand that the combinatorial patterns can be largely categorized to 11 clusters in 2014. In particular, we can see undetected relationships among different branches in the tree diagram in the interlaced colors in the segment of the color bar. For example, in 2010, dark blue color bars are interlaced with a light blue color, as they ultimately have a nested hierarchical relationship. They share the same APIs including both internal and external APIs.

Third, from the color-coded landscape map, we can visually understand that numerous plug-ins were generated in a short period of time. The landscape map can be understood from two different regions. In the diagonal line region, we can see how many different clusters are generated with different structural patterns. Each cluster in a diagonal line represents an underlying structure generated without a central designer. Thus, the analyzed results show that clusters are not generated in a uniform way. The organized generation of clusters indicates how many different types of plug-ins were generated without the control of a platform owner. However, this analysis provides an insight that the generation of plug-ins has a certain structure based on combinatorial patterns from the hierarchical order. The off-diagonal region shows the interdependency of each cluster specified by a color bar. There are red color-coded regions in the off diagonal region which change over time. The change implies that clusters are interconnected with each other in terms of the usage of APIs. The degree of interaction across clusters increases in a network over time. We cannot detect any uniform pattern over time, which suggests that the interdependency has an arbitrary relationship.



143 plug-ins in Dec 2006	4,770 plug-ins in Dec 2010	23, 985 plug-ins in Dec 2014

Figure 2. The evolution of a plug-in network

From the analysis, we can visually and statistically detect that a digital ecosystem does not have a certain structure but changes over time. Even though the analysis of a co-expression network offers an insight to understand how digital innovation evolves over time, some limitations exist to specifically understand the impact of combinatorial patterns on plug-in generation. Thus, we extract the plug-in information and API information with respect to a specified color bar over time and run a statistical analysis based on a negative binomial regression model.

Econometric Analysis

After the network analysis from 2004 to 2014, we extracted monthly data from 2004 to 2014 for the variable of statistical models in each cluster in the analyzed co-expression network. Table 1 indicates the descriptive statistics. The two main variables (the size of clusters and the size of sub-clusters) are extracted from the network analysis. The control variables include the age of plugins, the number of developers per plugins to understand the role of third party developers, and the number of plug-in update to see the role of improved plug-in quality on the evolution of a digital ecosystem. In particular, for the list of sub-cluster core and periphery APIs, we used the core/periphery technique (Borgatti and Everett 2000) to categorize APIs into three different types: universal core, sub-cluster, and periphery. First, we transformed the plug-in by API format to the edge-list format that shows the source and target node in a network. We considered the weighted value of source and target API connections by counting how many times the connection of two APIs is repetitively used in all plug-ins in each month. To categorize the universal core APIs, we considered the weighted value which represents how strongly the two APIs are connected in a network in the core/periphery analysis to capture the functional significance of APIs. We applied the core/periphery analysis in all edge-lists from January 2004 to December 2014. In January 2004, there were 4 universal core APIs. In December 2014, 12 APIs were used as universal core APIs. Second, we did another core/periphery analysis for sub-cluster and periphery APIs depending on each cluster extracted from the network analysis. We used the plug-in list of clusters from the co-expression analysis after extracting the list of universal core APIs in the edge-list in each cluster in each month. We could get the broader range of sub-cluster APIs without considering the weighted value. This is to reduce the emphasized role of internal APIs whose some of functions are required to be included in plug-ins to be uploaded in the WordPress system. In January 2004, there were 10 sub-cluster APIs including 3 external APIs in a cluster. In December 2014, there were 120 sub-cluster APIs that included 39 external APIs in total. Finally, the other APIs after extracting sub-cluster APIs were considered periphery APIs

Variables	Obs	Mean	Std.Dev	Min	Max
Number of nested hierarchies (t)	433	8.665	4.73	2	29
Number of plug-ins (t-1)	422	1462.436	844.5282	86	3758
Sub-cluster APIs (t-2)	411	27.836	24.554	1	118
Periphery APIs (t-2)	411	80.082	37.005	9	148
External API ratio in sub-cluster APIs (t-1)	422	5.597	7.651	0	37.5
Version Upgrade (t-1)	422	.038	.026	0	.187
Plugin/Developer	422	1.796	.16	1.373	2.244
Plugin Age	422	59.551	28.591	14.049	133.093

Table 1. Descriptive Statistics

	Model 1 (DV: Plugins (t-1))	**Model 2** (DV: Hierarchy (t))	**Model 3** (DV: Hierarchy (t))
Plugins (t-1)	-	.007(.0006)***	.004(.0006)***
External API ratio (t-1)	-	.239(.05)***	.039(.054)
Plugins x External API ratio (t-1)	-	-.0001 (.0001)***	-.0001(.0001)**
Plugin/Developer (t-1)	-	-2.739 (1.415)*	-1.417(1.585)
Plugin Age (t-1)	-	-.114(.039)***	-.029(.035)
Version Upgrade (t-1)	-	0.004 (.006)	-.007(.006)
Sub-cluster API (t-2)	-3.938(2.614)	-	.026(.026)
Periphery API (t-2)	12.252(3.83)***	-	.322(.036)***
$\text{Sub}-\text{Cluster API }(t-2)^2$	.103(.021)	-	.0001(.0002)
$\text{Periphery API }(t-2)^2$	-.06(.02)***	-	-.002(.0002)***
Plugin/Developer (t-2)	628.998(163.393)***	-	-
Plugin Age (t-2)	8.944(3.806)**	-	-
Version Upgrade (t-2)	1.449(.722)**	-	-
Constant	-2486.297(703.322)***	17.865 (7.162)***	.246(6.772)
N	411	422	411
R^2	.696	.52	.575

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

Table 2. OLS with Cluster and Time Fixed Effect Model

Table 2 presents the results of the model in Figure 2. Model 1 shows the role of APIs in sub-cluster and periphery sections on the generation of plugins. The increased number of APIs in the sub-cluster section does not significantly affect the new generation of plug-ins. This result implies that APIs in the sub-cluster section show a limited role of actively increasing the number of plug-ins. In addition, the square value does not show significance that APIs in the sub-cluster section do not continuously increase from the periphery section over time. Thus, H1a and H1b are not supported. However, APIs in the periphery section show the positive trajectory and non-linear shape of growth of the number of plug-ins over time. The increased number of APIs in the periphery section positively affects the growing number of plugins. Thus, H2a is supported. The square values of periphery APIs represent negative coefficients with significance at the 0.01 level. The negative sign implies that the rate of change decreases as the number of APIs increases. Thus, H2b is supported.

In model 2, we tested the role of external APIs in sub-cluster sections on the change of the size of sub-clusters that represent the increased functional heterogeneity. The model shows a positive role of previous plugins on forming the sub-clusters (or nested hierarchy) at the 0.01 level, as the increased number of plug-ins shows diverse functions. The role of external APIs in sub-cluster section positively affects the growth in the nested hierarchy at the 0.05 level. Thus, H3a is supported. The magnitude of coefficients indicates that external APIs in the sub-cluster section highly contribute to the generation of nested hierarchy than the increased number of plugins. In order to understand the specific role of external APIs in the sub-cluster section, we built the interaction term that shows negative signs at 0.01 level and is quite precisely estimated with a standard error of 0.0003. The negative sign represents that not all external APIs largely contributed on the increase of nested hierarchy. There are few external APIs highly contributed to increase the size of heterogeneity.

Third, model 3 predicts the overall impact of variables in different time periods on the growth of sub-clusters over time. From the model, we can specifically explore the role of external APIs in the sub-cluster section to estimate and predict the evolution pattern over input variables. The increased ratio of external APIs does not show significance. Thus, the result shows the limited role of external APIs in the sub-cluster section once the network structure is constructed. Thus, H3a is not supported in model 3. However, the interaction term still shows significance at the 0.05 level. The negative coefficient represents that 1) the role of external APIs is limited in forming the skeleton of a network and 2) few external APIs have a high impact on increasing the size of sub-clusters. The functional uniqueness of external APIs can contribute to change the structural pattern of change over time. Thus, H3b is supported in model 3. Periphery APIs represent external APIs in most of the time. From the result, periphery APIs can have their own topological structure over time depending on the diverse functions. Thus, the increased number of periphery APIs affects the topological structure. However, the negative coefficient of the square value of periphery APIs implies that the influx of new external APIs does not guarantee that the rate of forming new topological structures would always increase over time. Thus, H2a and H2b are still supported in model 3.

One interesting finding is the limited role of developers. We only included the other input variables at time t-1. In model 2, developers who are actively making plugins contribute to change the evolution pattern. The negative sign implies that some developers actively contribute to generate new plugins, while the others create limited numbers of plugins. The input variable shows significance at the 0.1 level. However, in model 3, the result shows insignificance when we test the role of APIs on the change at the same time. Thus, the result indicates that the role of APIs is stronger than the role of developers in the evolution pattern. The developers' role on the generation of new plugins depends on the various functional types of APIs.

Discussion

This paper explores the evolutionary pattern of digital innovation in a digital ecosystem that does not have a centralized control by a platform owner. Such a digital platform is highly dynamic and generative as it often invites third-party developers to create digital products that often go beyond the original design intent of the platform owner. To capture the evolution of the structure of the platform, we adopt an evolutionary network perspective by focusing on the evolution of the topological structures of a bipartie network derived from the combination of digital components. Our results complement previous studies on system design and innovation (Baldwin and Clark 2000) in a modular architecture. In particular, the analysis of a combinable developmental process based on generativity explains how digital components interact with other components across different functional groups to create new digital products. The unique structural patterns in a digital ecosystem are different from the structure of decomposable systems in a modular architecture.

An evolutionary network perspective provides a theoretical and methodological lens to explore the dynamic pattern of the ever-changing landscape of a digital ecosystem. First, the model allows us to think of two different roles of components on local optimum in an ecosystem characterized as a "landscape" (Kauffman 1993). Components working as "basins of attractors" determine the degree and pattern of changes, while other components follow the change led by basins of attractors for local optimum in a landscape (Levinthal 1997). To understand the structural impact of basins of attractors on evolutionary patterns, we consider the topological structure of clusters in a network based on genetic "modularity" (Wagner et al. 2007).

Our study is the first empirical study that has identified the basic evolutionary pattern of a digital ecosystem and how the infusion of new digital components affects the structural change of the ecosystem. We were able to demonstrate how new clusters of digital products in a digital ecosystem emerge and divide over time. Specifically, we found that the universal core – the APIs that are most frequently used by all plug-ins – actually do not contribute to the growth of the ecosystem. This is surprising as the platform owner controls all of those APIs in the universal core. To the contrary, the number of cluster core components and the number of periphery components influence both the growth of a cluster as well as the sub-division of a cluster. While the cluster core components include some external APIs, the periphery components are all external. Furthermore, the external cluster core components have a relatively greater impact on the sub-division of clusters, compared to internal cluster core components. Taken together, while platform owners certainly play an important role in building vibrant digital ecosystems, they alone cannot make the ecosystems grow. Without significant infusions of foreign elements that resides the boundary of the platform owner's control, the ecosystem may not grow as dynamically as it does with them. Particularly interesting to observe is the role of external APIs as part of cluster core components. These cluster core components are the ones that create the functional diversities across clusters. In the WordPress ecosystem, one can conclude that external APIs played a significant role in the growth of the ecosystem both in size and diversity over time.

Generativity (Zittrain 2006) is a unique concept to explain digital ecosystems with a layered modular architecture (Yoo et al. 2010a). This paper contributes to the understanding of the generativity of a digital ecosystem by focusing on the topological structure in a network to capture the evolutionary pattern that occurs without the central control of a platform owner. This paper explores the dynamic mechanism of digital innovation, asking how combinatorial innovation takes place with digital components in a digital ecosystem. In particular, we capture the non-linear interaction of combinatorial patterns at a certain point in time when the rate of digital innovation can be decreased even though the number of digital components continuously increases. This finding leads us to think about the role of a digital ecosystem after software-based platform innovation begins to show a certain evolutionary pattern. In particular, the result allows us to think about the role of third party developers who closely depend on the diverse types of digital components on the generation of functionally heterogeneous digital products.

Even though this paper makes theoretical and methodological contributions in several disciplines, there are a number of limitations to this study. Above all, this paper captures the evolutionary pattern of a focal platform from a technological perspective by focusing on a structural aspect in a network. One of the most valuable aspects of a digital ecosystem is that we can see third-party developers' information on digital products to find how their behavioral patterns influence one another. If there are unobserved behavioral patterns, we can specifically explore the reason why the rate of digital innovation decreases when the possible number of combinations increases. In addition, we did not take into consideration the market demand for digital innovation because of data limitations. Previous studies argue that an innovation does not emerge spontaneously but is a result of market demand (Clark 1985). Platform service providers open their platform information to the public because they are not able to satisfy users' demands on their own.

This paper provides a useful insight to understand the evolutionary pattern of digital innovation using the logic of generativity in a more systematic way. By exploring how the structural pattern changes over time, our findings provide a way to think about how continuous digital innovation can occur in a digital ecosystem.

Conclusion

As the strategic and economic importance of digital ecosystems evolve over time, so does the importance of a theoretical understanding of how such evolutions take place. Our study offers a new perspective to examine such evolutionary dynamics and the specific mechanisms that produce them. Even though this study is limited by its focusing on a single ecosystem, we hope that we have taken an initial step to explicate the dynamic nature of generativity. At the same time, we hope to expand the scope of this study to understand this dynamic nature by considering more and different aspects.

References

Adomavicius, G., Bockstedt, J. C., Gupta, A., and Kauffman, R. J. 2008. "Making Sense of Technology Trends in the Information Technology Landscape: A Design Science Approach," *MIS Quarterly* (32:4), pp. 779-809.

Anderson, P., and Tushman, M. L. 1990. "Technological Discontinuities and Dominant Designs: A Cyclical Model of Technological Change," *Administrative Science Quarterly*, pp. 604-633.

Baldwin, C., and Von Hippel, E. 2010. "Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation," *Harvard Business School Finance Working Paper*:10-038), pp. 4764-4709.

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

Baldwin, C. Y., and Woodard, C. J. 2009. "The Architecture of Platforms: A Unified View," *Platforms, Markets and Innovation*, pp. 19-44.

Barabási, A. L., and Oltvai, Z. N. 2004. "Network Biology: Understanding the Cell's Functional Organization," *Nature Reviews Genetics* (5:2), pp. 101-113.

Benkler, Y. 2006. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale Univ Pr.

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.

Boudreau, K. J. 2012. "Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation," *Organization Science* (23:5), pp. 1409-1427.

Boudreau, K. J., and Jeppesen, L. B. 2014. "Unpaid Crowd Complementors: The Platform Network Effect Mirage," *Strategic Management Journal*.

Ceccagnoli, M., Forman, C., Huang, P., and Wu, D. 2012. "Cocreation of Value in a Platform Ecosystem: The Case of Enterprise Software," *MIS Quarterly* (36:1), pp. 263-290.

Chesbrough, H. W. 2003. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. Harvard Business Press.

Clark, K. B. 1985. "The Interaction of Design Hierarchies and Market Concepts in Technological Evolution," *Research Policy* (14:5), pp. 235-251.

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.

DeLanda, M. 2013. *Intensive Science and Virtual Philosophy*. A&C Black.

Dhar, V., Geva, T., Oestreicher-Singer, G., and Sundararajan, A. 2014. "Prediction in Economic Networks," *Information Systems Research* (25:2), pp. 264-284.

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

Fleming, L. 2001. "Recombinant Uncertainty in Technological Search," *Management Science* (47:1), pp. 117-132.

Fleming, L., King, C., and Juda, A. I. 2007a. "Small Worlds and Regional Innovation," *Organization Science* (18:6), pp. 938-954.

Fleming, L., Mingo, S., and Chen, D. 2007b. "Collaborative Brokerage, Generative Creativity, and Creative Success," *Administrative Science Quarterly* (52:3), pp. 443-475.

Fleming, L., and Sorenson, O. 2001. "Technology as a Complex Adaptive System: Evidence from Patent Data," *Research Policy* (30:7), pp. 1019-1039.

Fleming, L., and Sorenson, O. 2004. "Science as a Map in Technological Search," *Strategic Management Journal* (25:8-9), pp. 909-928.

Goldberg, D. E., Deb, K., Kargupta, H., and Harik, G. 1993. "Rapid, Accurate Optimization of Difficult Problems Using Messy Genetic Algorithms," *Proceedings of the Fifth International Conference on Genetic Algorithms,(Urbana, USA)*: Proceedings of the Fifth International Conference on Genetic Algorithms,(Urbana, USA), pp. 59-64.

Hausman, J. A., Hall, B. H., and Griliches, Z. 1984. "Econometric Models for Count Data with an Application to the Patents-R&D Relationship," National Bureau of Economic Research Cambridge, Mass., USA.

Henderson, R. M., and Clark, K. B. 1990. "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly*, pp. 9-30.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. U Michigan Press.

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

Kauffman, S. A. 1993. *The Origins of Order*. Oxford University Press New York.

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.

Langlois, R. N. 2002. "Modularity in Technology and Organization," *Journal of Economic Behavior & Organization* (49:1), pp. 19-37.

Levinthal, D. A. 1997. "Adaptation on Rugged Landscapes," *Management Science* (43:7), pp. 934-950.

Nelson, R. R., and Winter, S. G. 1982. *An Evolutionary Theory of Economic Change*. Belknap press.

Newman, M. E. J., Watts, D. J., and Strogatz, S. H. 2002. "Random Graph Models of Social Networks," *Proceedings of the National Academy of Sciences of the United States of America* (99:Suppl 1), pp. 2566-2572.

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.

Schilling, M. A. 2002. "Modularity in Multiple Disciplines," *Managing in the Modular Age: Architectures, Networks and Organizations*, pp. 203-214.

Schilling, M. A., and Phelps, C. C. 2007. "Interfirm Collaboration Networks: The Impact of Large-Scale Network Structure on Firm Innovation," *Management Science* (53:7), pp. 1113-1126.

Simon, H. A. 1962. "The Architecture of Complexity," *Proceedings of the American philosophical society*, pp. 467-482.

Strogatz, S. H. 2001. "Exploring Complex Networks," *Nature* (410:6825), pp. 268-276.

Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. 2003. "A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules," *Science* (302:5643), p. 249.

Tiwana, A. 2015. "Evolutionary Competition in Platform Ecosystems," *Information Systems Research*.

Tiwana, A., Konsynski, B., and Bush, A. A. 2010. "Platform Evolution: Coevolution of Platform Architecture, Governance, and Environmental Dynamics," *Information Systems Research* (21:4), pp. 675-687.

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.

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., 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.
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.
Zhang, B., and Horvath, S. 2005. "A General Framework for Weighted Gene Co-Expression Network Analysis," *Statistical Applications in Genetics and Molecular Biology* (4:1), p. 1128.
Zittrain, J. L. 2006. "The Generative Internet," *Harvard Law Review*, pp. 1974-2040.