Evolution of Business Ecosystems

Hirofumi Tatsumoto

Abstract The aim of this chapter is to investigate the dynamics of business ecosystems, which are a special form of industrial cluster characterized by diversity of firms, complex relationships among them, and the presence of platform firms. Such characteristics come from network effects. Firms often set open standards to trigger network effects that are advantageous for their business models. Among them, platform firms, which strategically use the open standards and make the most of the network effects in relation to their business models, play a special role in business ecosystems, because their strategic behavior contributes not only to increasing their market presence but also to the expansion of the ecosystem by stimulating the entry of newcomer firms.

To predict the dynamics of business ecosystems, we should try to understand how product architecture influences firm collaboration within them and characterizes their industrial structures. Open-modular systems, such as PCs, tend to modify their industries and turn them into business ecosystems, whereas closed-integral systems, such as automobiles, yield an industrial form called cohesive clique. Owing to the linkages between architecture and industrial cluster, changes in architecture lead to actual changes in industrial clusters.

Two main factors have recently been affecting product architecture, i.e., digitalization and globalization. Digitalization works as a trigger for architectural change, and globalization acts as its amplifier, which means that products whose architectures have seemed stable for years might drastically change. A slight change in architecture often causes the industrial structure to quickly transform from a cohesive clique into a business ecosystem, which naturally causes changes in the profitability and competitiveness of firms. Faced with these architectural dynamics, each firm needs to thoroughly understand the connections between product architecture and industrial cluster.

H. Tatsumoto (✉)
Graduate School of Business Sciences, University of Tsukuba, Tsukuba, Japan
e-mail: tatsumoto@gssm.otsuka.tsukuba.ac.jp

T. Fujimoto, F. Ikuine (eds.), *Industrial Competitiveness and Design Evolution*,
Evolutionary Economics and Social Complexity Science 12,
https://doi.org/10.1007/978-4-431-55145-4_5

1 Business Ecosystems

1.1 What Is a Business Ecosystem?

The present book has so far analyzed the evolution of product designs and organizational capabilities, as well as their impact on product competitiveness and demand creation, assuming that such products are stand-alone and compete with other products. This chapter expands the research scope to analyze the evolution of industries in which certain products complement other products. The relationship among products is more complex in this case, since they interact not only by competing with but also by complementing one another.

When we find both competition and complementation in an industry or industrial cluster, we call the said industrial system a business ecosystem. Thus, a business ecosystem is an industrial cluster characterized by multiple products and by network effects among them. A product can be complementary to others, which means that its growth depends not only on itself but also on other products. For example, if sales of hardware increase, sales of software will increase too, as software is a complementary good to hardware. This mutual growth is the result of network effects among products.

A business ecosystem contains numerous network effects within it. Indeed, network effect is a significant strategic factor, since an additional user of a product has a positive effect on the value of that product to others. Thanks to the presence of network effects, the value of a product increases according to the number of users. The users benefit not only from the product itself but also from the number of other users, and, in some cases, they value the number of other users more than the product itself. The telephone is a classic example, as a telephone user enjoys greater benefits when he/she belongs to a large telephone network.

What makes the situation more complex is the fact that network effects can emerge from complementary goods. In the example above, when a user chooses a smartphone, he/she should consider not only the number of telephone users but also the variety of apps. Apps are typical complementary goods for smartphones, and their variety is often referred to as an indirect network effect, whereas the number of telephone users is referred to as a direct network effect. Both network effects affect user preferences and the business of firms.

Moreover, the evolution of a business ecosystem is different from that of an ordinary industrial cluster. One major difference is the presence of platform firms, which play leading roles in the evolution of the business ecosystem itself. Previous research has paid much attention to these firms and given them several names: keystone firms (Iansiti and Levien 2004a), platform leaders (Gawer and Cusumano 2002), platform providers, and technology enablers. The increasing number of studies on the subject indicates that there is considerable interest in their special role and overwhelming performance in business ecosystems.

Here, we will focus on the evolution of business ecosystems, with special attention to their dynamics, which are affected by architectural change. This chapter is organized as follows. The first section offers a brief overview of the concept of business ecosystem and illustrates its dynamics and growth model. The second section deals with architecture, which is the main subject of this book, and with firms' approaches to design problem-solving. The linkage between architecture and firms' behavior provides the theoretical basis for analyzing the impact of architectural change on business ecosystems. The third section first explores the recent developments of digitalization and globalization and then incorporates them into the analysis. Additionally, it discusses architectural change and its impact on business ecosystems by drawing a comparison between automobiles and personal computers. The fourth, and last, section presents some concluding remarks.

1.2 An Industry with a Layered Structure

The concept of business ecosystem dates back to the 1990s. Grove (1996) was the first person who identified the structural change occurring in the computer industry. In his book, Grove described this change as "vertical silo to horizontal layers." Before the mid-1990s, computer firms had their own operation chains based on the industrial structure of computers. IBM had its own chips, components, assembly manufacturers, and retailers, and its operation chain was dedicated only to the firm itself. Thus, operation chains were closed networks in those days. Yet, a major architectural change happened in computer systems in the mid-1990s, due to the shift from mainframes to personal computers. The architecture of computers became open, and manufacturers freely chose chips or components from suppliers to produce their goods. Computer retailers sold any brand of computer products. There were no longer closed relationships in the industry.

This architectural change of computer systems also caused a structural change in the computer industry. The operation chains became more open and firms flexibly connected or disconnected with other firms. Computer makers could easily modify their operation chains because computer systems now had open interfaces that enabled compatibility across many components. This open flexibility changed the industrial structure from vertical silo to layer by layer, with each layer standing for chipmakers, component providers, assembly manufacturers, or retailers. The layers had open relationship based on open standards. This open, layer-by-layer structure is the origin of the concept of business ecosystem.

Such new form of industrial structure originally emerged in personal computers and then prevailed in digital consumer goods, cell phones, and similar products. The Internet significantly accelerated this move, and all the industries related to the Internet came to form the same business ecosystem.

1.3 Analogy with Biology: Diversity, Symbiosis, and Keystone

The term *business ecosystem*, the definition that Iansiti and Levien introduced into the academic literature to describe this industrial structure, comes from biology (Iansiti and Levien 2004b), since analogies are drawn with biological ecosystems. Indeed, business ecosystems share some common features with biological ecosystems.

The first common feature is diversity. A biological ecosystem comprises many species that interact with one another in the natural environment. For example, the African savannah ecosystem includes a diverse community of species, such as tropical grasslands and various animals. Similarly, a business ecosystem is an industrial cluster comprising a diverse community of firms. Each firm plays its role by providing components, developing products, and delivering them to users.

Secondly, business ecosystems have complex relationships. Organisms in biological ecosystems interact both directly and indirectly. The food chain typically involves direct relationships, but there are also indirect relationships, like symbiosis. The case of sea anemones and clownfish is famous for its mutualistic relationship of symbiosis. Sea anemones provide clownfish with a safe home, and, in return, clownfish clean the anemone and offer nutrients in the form of waste. Firms in business ecosystems have relationships that are similar to those found in biological ecosystems. They interact directly with one another along the supply chain for parts/materials or services, but they also interact indirectly by means of compatible standards. When compatible standards are present, the sales of a product affect those of compatible or complementary products. Even when firms have no direct mutual relationships, i.e., buying or selling, they need to be aware of this indirect relationship. Indirect relationships in business ecosystems are referred to as network effects.

The third characteristic is the presence of special actors within the ecosystem. In biological ecosystems, some species have an outsized impact on the ecosystem. Paine, a famous ecologist studying biological diversity, conducted field research and found that removing a specific species from an environment can severely affect neighboring species (Paine 1966). He called these special species keystone species. Business ecosystems are much the same, in that they also comprise special firms. Iansiti and Levien called those special firms keystone firms (Iansiti and Levien 2004a). The keystone firms that they mentioned included many platform providers, such as Microsoft and Qualcomm, which are typically called platform firms—and this is the terminology that we will adopt in the remained of our analysis.

Many economists and management scholars have studied platform firms and their business strategies, because they often have dominant market shares and affect neighboring firms. National authorities investigate platform firms in relation to potential breaches of antitrust laws, economists want to theorize their business models in light of their economic behavior, and management scholars wish to understand the secret of their success.

In brief, business ecosystems are characterized by diversity of firms, complex relationships among them, and the presence of platform firms as special firms. These concepts, based on the analogy with biological ecosystems, are useful to describe this new form of industrial structure. Network effects, as an invisible engine, are the key aspect behind these characteristics of business ecosystems (Evans et al. 2006). Firms use network effects to deliver new value, to enlarge their market, or to gain competitive advantages.

When network effects are extended to more and more products or services, the industrial structure is likely to take on the form of a business ecosystem. As mentioned above, business ecosystems emerged first in the personal computer industry, followed by digital consumer goods and cell phones. The Internet strongly reinforced this shift. Today, business ecosystems are found in the machine tool and automotive industries too, which are usually regarded as legacy industries. Hence, this structural change has occurred not only in the new digital industry but also in traditional ones.

1.4 Open Standards as a Source of Network Effects

A business ecosystem is an industrial structure with network effects, which come from complementary relationships among products or services. In many cases, system products have complementary goods. For example, personal computers consist of hardware and software. Software and hardware have a complementary relationship, since for hardware makers software is a complementary good and vice versa.

System products and complementary relationships can easily be found in everyday life. DVD software and DVD players, smartphones and apps, and TV and TV programs are some typical cases. Even the automobile business, which is a very traditional industry, entails complementary relationships. Since system products have become common in our daily life, network effects are ubiquitous, and firms try to exploit them to boost their business.

Whether a business is new or old has nothing to do with the concept of business ecosystem. Complementary relationships are the key factor here, because they generate network effects. The things that one counts within a system determine the boundary of its business ecosystem, so the idea of business ecosystem is deeply rooted in how one sees related workings and recognizes them as a system. Through a broader lens, even a traditional industry can be regarded as a business ecosystem.

Let us look, for example, at the automobile business, which is a traditional industry. If we focus only on automakers and suppliers, it does not appear to have the industrial form of a business ecosystem, because there are only direct relationships and no complementary relationships along the supply chain. But, once we broaden the scope, we can indeed view the automobile industry as a business ecosystem. Within the whole transportation system, automobiles have a complementary relationship with the energy business. Refueling periodically is necessary to drive a vehicle, so that automobiles and gasoline are characterized by complementarity and network effects between them.

All the examples above share the common denominator of compatible standards among products. PCs and smartphones have APIs for developing software and apps, DVDs have various formats for developing content, and automobiles have clear gasoline quality standards to guarantee compatibility. Firms within each ecosystem share these compatible standards, which are called open standards because they are openly shared by the firms in the industry.

Open standards are a powerful source of network effects, since firms need them to collaborate with other firms. Collaboration is a key factor to address the increasing complexity of the system, due to the fact that no single firm can deal with the whole system alone. Open standards provide the pivotal basis for firm collaboration and eventually generate network effects.

Business ecosystems contain many open standards as a source of network effects, but these do not necessarily appear as formal standards. They can take on many forms, such as roadmaps for investments, open APIs, open-source software, or protocol documents. All of them work as open standards because firms openly share them and collaborate based on them, but it does not matter whether the standards are formal or not. Even when they do not entail penalties for protocol violation, they still work well as open standards. As a source of network effects, it is irrelevant whether the open standards are compulsory or voluntary, and firms often follow them by collaborating in an autonomous, decentralized manner.

Business ecosystems offer firms direct and indirect ways to interact with other firms. Supply chains are a direct means of interaction, which is typical in many industries. The use of open standards is an indirect form of interaction that enables firms to collaborate in an autonomous, decentralized manner and is specific to the case of business ecosystems. The notable fact is that open standards generate the network effects that characterize the dynamics of business ecosystems.

1.5 Emergence of Platform Firms

Network effects characterize the behavior of firms in a business ecosystem. Firms have different ways of exploiting network effects depending on their business models, and their strategic behavior is mainly divided into three types: designing of network effects, matching of two markets, and bundling a set of goods.

These strategic actions put in place by firms can contribute to their competitiveness. Although exploiting network effects is common to all the firms in a business ecosystem, platform firms, which play a special role, are in a position to make the most of network effects and usually pursue all three actions above.

The first action is the designing of network effects. As they strongly impact on business growth and competitiveness, designing desirable network effects is a basic strategy. As discussed above, open standards are a powerful source of network effects. Consequently, firms often set open standards to trigger network effects that are advantageous for their business models. There are three approaches to setting open standards, which can be de facto, de jure, or consensus standards.

De facto and de jure standards are more traditional forms. De facto standards are automatically defined through market share dominance, while de jure standards are established by public committees, such as ANSI and ISO/IEC. As a new approach, consensus standards are set by firms gathering together in consortia, forums, or alliances. They are thus similar to de jure standards but differ in the fact that firms come together according to their own will, so multiple consortia might be created to pursue similar targets. We will discuss the differences among these three approaches to standards setting later on in this chapter. Platform firms are likely to organize the consortia tasked with setting open standards. Their purpose is not only to establish a basis for collaboration but also to try and make desirable network effects emerge in the ecosystem.

The second strategic action, the matching of two markets, is another typical firm approach aimed at exploiting network effects. For example, a firm develops a navigation app for car drivers and collects information on where they go or what they are likely to do. Based on this information, the firm is able to suggest a restaurant or shopping mall where the drivers might want to stop. In this case, the navigation firm acts not only by giving information on the goal but also by matching two markets, the navigation service and the other services that the car drivers might want to receive according to their preferences. Matching works well if the two products are linked by a network effect, since consumption of one good leads to increased consumption of the other good. The network effect between two products means that the user of a product is likely to benefit from the other product when he/she jointly uses them.

Scholars have studied matching as the theory of two-sided markets or pricing (Rochet and Tirole 2003). Thanks to network effects, matching between two markets contributes to the expansion of the total market, ultimately supporting the growth of the business ecosystem as a whole. Of course, this also leads to an increasing presence of firms that provide matching services, which are often referred to as platform firms. So platform firms play a special role in business ecosystems.

Firms often use matching as a critical part of their business models, especially in Internet services. Google, Facebook, and Amazon are the technology giants that provide matching services. These platform firms work as matching agents between two or more markets by offering open interfaces for third parties to use their services. The third parties use the open interfaces, collect information about the users, and reinforce the network effect by matching. Real businesses also use matching of two groups. For example, in the area of the Internet of Things, or IoT, there are many IoT devices that generate data about the usage of a product/service. Matching in real business connects these usage data with other products/services. As seen above in the case of a car navigation app, matching connects the navigation service with the restaurant or shopping service.

The third way to make the most of network effects is the bundling of a set of goods or services as a package (Nalebuff 2004; Eisenmann et al. 2011). Both matching and bundling rely on network effects between two groups, but they differ in the way in which the two groups overlap. Matching works well if the two groups do not have any overlaps. For example, if the providers and consumers of a given product are different groups and they do not simultaneously produce or consume the said good, firms can exploit the network effects by matching. But if the two groups mostly overlap—for example, in the case of cell phones and portable audio devices—matching is rather useless, while bundling is an effective strategy to make the most of network effects. Smartphones, which bundle together cell phone and portable audio features, are a good choice for users in situations of considerable overlaps.

If two products are functionally complementary, there will be network effects between the two groups of users. And if there are major overlaps between the said two groups, most users will benefit from the two functions combined. With functional complementarity, meaning the network effect between two products, the utility of a package bundling two functions together exceeds the sum of the utility of the two individual functions. The utility gap between the package and the sum of individual goods widens as functional complementarity becomes stronger.

Bundling works as a kind of lock-in or closure strategy. The firms that bundle two functions in a package enclose the network effects into their products, so that they can benefit from them while other firms do not. For instance, smartphone manufacturers lock in consumers who use cell phones and portable audio. Functional complementarity makes it much harder for firms able to provide one function only to enter the smartphone market. Since network effects derive from functional complementarity between two goods and business ecosystems are characterized by considerable network effects among products, firms have many opportunities to implement bundling strategies.

The three strategic actions described above are quite different from those of product firms operating in ordinary markets, whose main aim is to provide good-quality products at inexpensive prices. Of course, firms in business ecosystems have the same aim, but they cannot disregard the new factor, i.e., network effects for creating value and competing with rivals, which represent a major challenge as well as an opportunity.

In sum, designing of network effects, matching of two markets, and bundling of goods are all strategic actions that exploit the macrostructure of business ecosystems. Since business ecosystems comprise a variety of products that complement each other, network effects will arise as an intrinsic feature that firms can leverage when implementing their strategic actions, which exceed the scope of the strategies that product firms typically have.

These strategic actions are very characteristic of business ecosystems, and the organizations that wish to implement them need strong macroscopic capabilities. With respect to this point, platform firms are able to make the most of these strategies as they do possess macroscopic capabilities. They concern themselves not only with their own products but also with neighboring products, capturing complementary relationships that work as network effects. They also watch the macroelements, such as consortia and open standards, that might be a source of network effects and carefully assess business ecosystem dynamics because these invisible forces affect

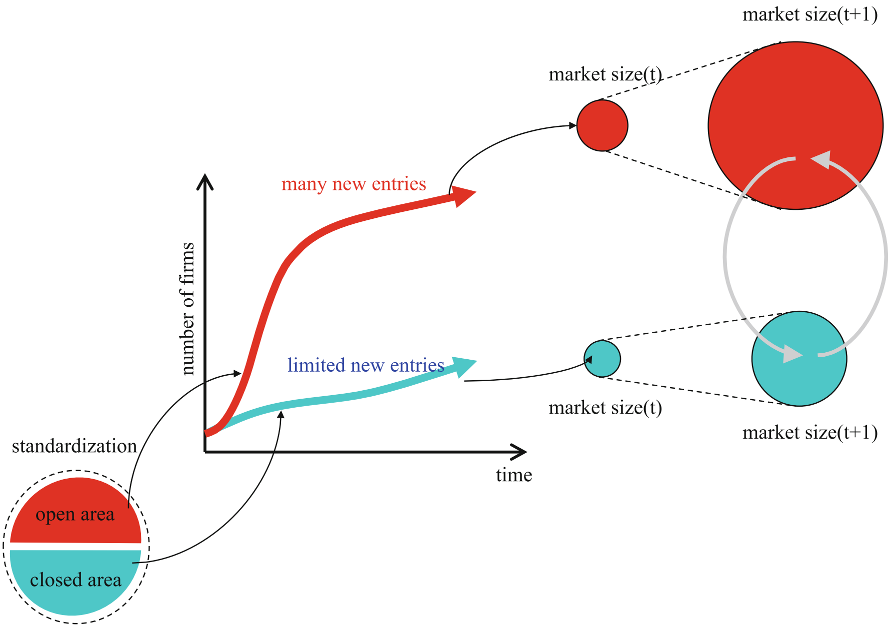

Fig. 1 Growth model of business ecosystem

their business. So, platform firms carefully monitor the dynamics of business ecosystems and exercise strategic actions to exploit them according to their business models.

1.6 Growth Model of Business Ecosystems

As discuss so far, a business ecosystem is an industrial cluster characterized by diversity of firms, complex relationships, and platform firms. Such characteristics come from network effects. Many market segments have indirect relationships that generate network effects, invisible forces affecting the evolution of a business ecosystem. Figure 1 illustrates the growth model of business ecosystems. For the sake of simplicity, we pick two segments that are related by network effects. If one segment grows, the other segment also grows due to network effects. Strong network effects cause a steeper mutual growth of the two segments.

If platform firms, which exploit network effects for their strategic purposes, find that two segments are linked by network effects, they try to leverage this in the most effective way. Typically, they start by setting their strategic framework, picking one of the two segments as the target segment and the other as a companion segment. They sell their own products in the target segment, which they want to grow in order to expand their business. One possible strategy is discounting, but applying cheaper prices is not a good idea even if it stimulates growth, since it can cause price erosion in the target segment and investments cannot be recouped. The platform firms expect the target segment to become the main source of profit for their business, so they decide to exploit the companion segment instead.

The companion segment, which comprises complementary products, is linked with the target segment by means of network effects. So, platform firms stimulate the companion segment so that the target segment can grow and they can recoup their investments. Since they do not discount in the target segment, they are free from the risks of price erosion and damage to their source of profit.

Multiple strategic actions are able to stimulate the growth of companion segments and reduce entry costs for newcomer firms, among which, for example, provision of development references, support to open-source software, and free licensing of patents. The most notable of these actions is the setting of open standards. Open standards emerge from the industry-wide sharing of technological information and irreversibly impact on the ecosystem because they can trigger massive entry of newcomers in the companion segment.

Newcomer firms are not familiar with the technological knowledge and market context of the companion segment. Open standards work as a good point of reference for any newcomers in terms of technological knowledge. In addition, they validate the quality of the products that the newcomer firms produce. Since open standards include explicit compatibility criteria, new entrants can easily adjust their products. Thus, the adoption of open standards compensates for the lack of newcomers' knowledge of the market context.

By learning from open standards, the newcomers rapidly catch up with the incumbent firms in the companion segment. The said segment is thus transformed into an open area for newcomers that would otherwise hesitate to enter. In brief, open standards remove the entry barriers existing in the companion segment. With new entries, market growth in the companion segment is stimulated, supply volumes increase, and prices drop to affordable levels. The size of the target segment increases depending on the number of newcomers entering the companion segment.

The growth of the companion segment also promotes that of the target segment. The network effect, which links the two segments and stems from their complementary relationship, conveys the growth momentum of the companion segment to the target segment. Hence, open standards make both the target and companion segment grow. A noticeable consequence of this process is that the characteristics of the two segments change. The companion segment, for which open standards have been set by the platform firms, becomes an open area for newcomer firms, as entry barriers are removed and opportunities to catch up multiply. Conversely, the target segment remains almost completely out of bounds to newcomers, since entry barrier are still in place. The main reason for this is the fact that platform firms, for which the target segment is the main source of profit, see any newcomers as rivals and strive to keep them out.

The setting of open standards causes business ecosystems to have an open area and a closed area. Newcomer firms enter the open area and stimulate its growth. Due to such increasing entries, the closed area, which is still mostly off limits to newcomers, also grows thanks to the network effect linking the two segments. Therefore, the moment when open standards are set often represents the inflection point at which a business ecosystem opens up one of its segments to newcomer firms and consequently starts to expand.

This is a typical strategy for platform firms because it allows them to achieve two key targets, i.e., ensuring the growth of the business ecosystem as a whole and keeping their own business profitable (Tatsumoto 2017). This so-called platform strategy can be seen in various ecosystems, such as personal computers, digital consumer goods, cell phones, and semiconductor industries, and even traditional industry firms have recently started to express an interest in it. The trend is becoming more and more evident because the industrial structures of various areas are transforming into business ecosystems. For example, in the automobile industry, this trend is often referred to as CASE, which stands for connected, autonomous, shared, and electric. In the case of machine tools, it is called Industry 4.0.

Before exploring this topic further, the next section will illustrate the theoretical framework adopted to analyze these trends, which is mainly based on the architectural theory.

2 Architecture

2.1 Architecture Affects Business Ecosystems

The first section of this chapter provided an overview of the dynamics of business ecosystems. Let us now shed light on the characteristics of business ecosystems, which are deeply connected with the architecture of the products or services provided by firms within them. The concept of architecture refers to the basic configuration design of an artifact, which depends on problem-solving decisions regarding its design factors.

This definition means that the architecture of a product is linked with the way in which a firm solves design problems to manufacture it. In particular, when dealing with large-scale design problems, which require collaboration among multiple firms, product architecture affects the activity of joint problem-solving. Some types of architecture are more suited to a small group of firms jointly dealing with design problems, whereas other types call for a more autonomous, decentralized approach.

With respect to design problems, each architecture has a different approach to developing design solutions. Increasing complexity reinforces this tendency and causes architectural changes, which impact on the industrial structure itself. Business ecosystems, which often arise as a consequence of architectural changes, are a special form of industrial structure. In extreme cases, a slight change in architecture might lead the industrial structure to quickly transform into a business ecosystem, which naturally causes changes in the profitability and competitiveness of firms. Hence, most of the firms who operate within an ecosystem cannot ignore the architecture of its products and how it changes.

In the remaining part of this section, we will examine the architectural characteristics that influence the behavior of firms by laying out the theoretical dimensions of product architecture. We will categorize these characteristics based on the features of the interfaces for which firms have to collaborate with one another and then discuss how and why product architecture affects the shape of the industrial structure.

2.2 Architectural Anatomies of Complex Systems

The study of architecture is rooted in Simon's examination of complex systems (Simon 1969). He found that complex systems share the common feature of having numerous design factors that are mutually dependent. The said design factors and dependencies are the source of complexity that makes product development difficult.

A set of design factors that are mutually dependent can be seen as a design problem of simultaneous equations. During the development process, product designers look for a solution to the design problem. Obviously, as the number of factors and dependencies rises, it becomes ever more difficult to solve the design problem, until a complexity limit is reached and product designers are no longer able to find a solution.

To handle this increasing complexity, product designers encapsulate the design factors into modules (Aoki 2001). A module is an assembled chunk of design factors that are mutually dependent. This design technique is a kind of "divide and conquer" approach to complexity, and it is usually referred to as modularization. Through modularization, a complex system comes to have relatively more dependencies within modules and, at the same time, relatively fewer dependencies across modules. Thus, modularization reduces the complexity of design problems because it divides the large simultaneous equations into small independent ones, so that product designers can handle them individually.

The concept of modularization is quite important, especially in the practical development process, which is often affected by interferences ranging from a simple telephone call to economic turbulence. To reduce this *noise*, product designers use the modularization technique and divide the design factors into modules. A set of design factors and dependencies within a module is unaffected by external noise, and, if the said noise damages a module, the other modules will remain safe. In this way, only one module is lost, and the product designers just need to replace the damaged module with a new one.

Therefore, modularization causes a complex system to become a composite of modules, which Simon (1969) referred to as a nearly decomposable system (NDS), and its behavior can be approximately represented by the combination of its modules' behaviors. Modern products, which are likely to have numerous design factors for many functions, are often complex systems, and we can regard them as a composite of individual modules.

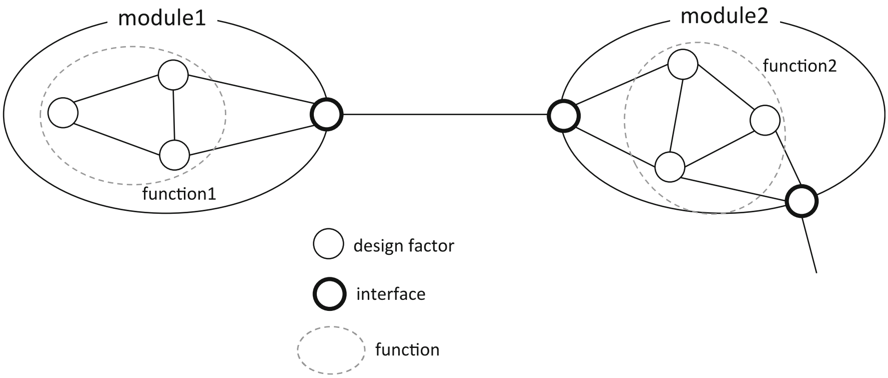

Fig. 2 Design factors, modules, and interfaces

2.3 Features of Module Interfaces

A module contains multiple design factors that realize its functions. Designing a function means solving simultaneous equations involving a set of design factors. Product designers need to identify appropriate parameters for all the design factors, and they usually adopt a trial and error approach to come up with solutions. Once a solution is found, the set of design factors will work well as a function of the system. That is, a set of design factors denotes a function in a module.

Figure 2 illustrates the concept of complex system as a composite of modules. Module 1 contains three design factors, which perform function 1. Similarly, module 2 consists of three design factors and fulfils function 2. Moreover, there are special design factors creating connections across the modules. These design factors, which are called interfaces, are gateways to the design factors within each module. The interfaces connect the modules and the composite of the modules makes up the system.

The combination of the individual modules' behaviors approximately represents the behavior of the system. From the users' point of view, behavior means function. In this interpretation, designing a complex product is an effort to combine modules. Yet, a further complication lies in the fact that the combination of modules produces interaction effects, which might appear negligible at first or immediately come across as difficult to eliminate. Even when they are initially weak, they might stack up and become collectively stronger. This is why those involved in designing complex systems tend to adopt a trial and error approach.

When combining modules, product designers use interfaces. Some interfaces might work well in combining modules, but they might not make the most of the modules' characteristics. Other interfaces might be difficult to use because they require information that is not shared. Hence, interfaces work as a constraint for design problems, and complex systems heavily rely on their characteristics.

Prior research on product architecture has shown that two features of interfaces particularly affect the designing activities. The first feature is how tightly or loosely the interfaces are able to couple the modules in the system with one another. The second feature is how openly the information of the interfaces is shared by the firms within the ecosystem. We will now explore these two features in greater detail.

2.4 Tightness of Interfaces

Let us turn to the first feature of interfaces, i.e., the ability to tightly or loosely couple the system modules, which plays a crucial role in determining whether a product is integral or modular. A complex system is composed of multiple modules. When the interface couples the modules tightly, connections are possible only among specific modules. By contrast, when the interface couples the modules loosely, they can easily connect with one another.

The reason for coupling modules is that a function in a module depends on another function in another module. Hence, product designers use interfaces for module coupling to connect a function to another function across modules. If a function depends solely on another function in the same module, the use of an interface is not necessary.

Based on this functional dependency across modules, the use of interfaces reflects the mapping pattern of functions and modules. Figure 3 shows the correspondence between interfaces and architecture.

In the left part of Fig. 3, (i) and (ii) show how the interface couples the modules: (i) refers to the case of tight coupling, whereas (ii) refers to the case of loose coupling of the modules.

In the right part of Fig. 3, (a) and (b) show the types of architecture through bipartite graphs illustrating how the modules and functions are connected. Prior studies have often used bipartite graphs to describe the mapping pattern of modules and functions in product architecture (Ulrich 1995). (A) describes a situation in which a function in a module heavily depends on other functions in other modules; this is the typical mapping pattern of integral architecture. Conversely, (B) describes a situation in which a function in a module does not heavily depend on other functions in other modules; this is the typical mapping pattern of modular architecture.

In the case of integral architecture, where functions heavily depend on one another across modules, a slight modification to function 1 affects both module 2 and module 3 exactly because of mutual dependencies. Since it is difficult to make changes, integral architecture cannot have many design options. Instead, when the architecture is modular, as in situation (B), slight modifications do not affect other modules, and design changes are easily implemented because mutual dependencies across modules are weak.

As for the relationship between interface and architecture, the arrows in Fig. 3 show that tight coupling corresponds to integral architecture and loose coupling to modular architecture.

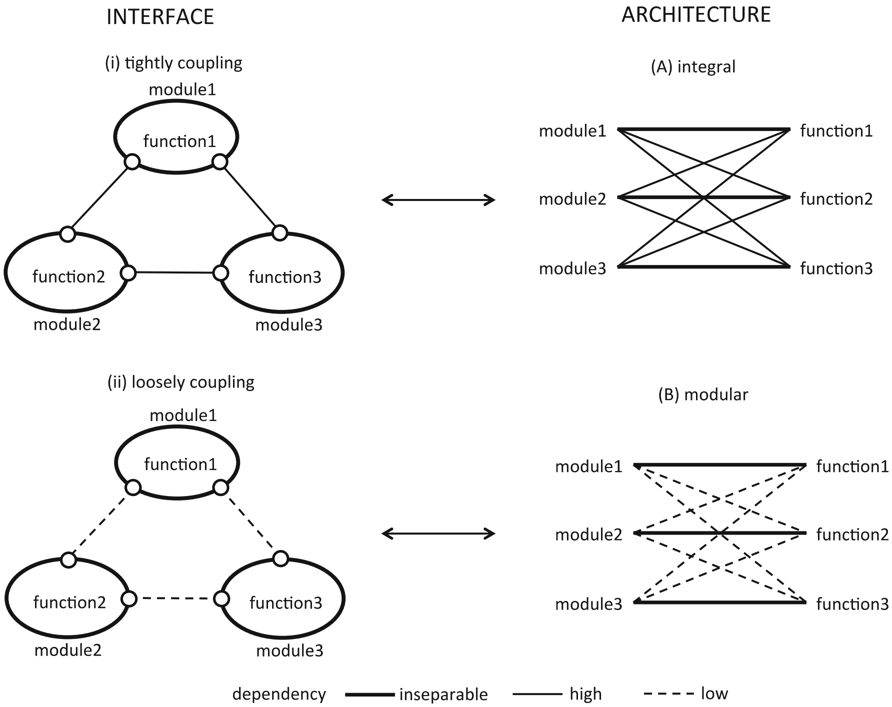

Fig. 3 Interfaces and architectures

In part (i) of Fig. 3, the interfaces tightly couple modules 1, 2, and 3. This means that function 1 in module 1 heavily depends on functions in modules 2 and 3. (A), the graph referring to integral architecture, expresses such dependency pattern in the form of mapping between functions and modules. Function 1 has inseparable mutual dependencies with module 1, which means that a set of design parameters of function 1 forms module 1. Function 1 also has strong dependencies with modules 2 and 3 because it depends on functions 2 and 3. These strong dependencies among functions cause the interfaces to tightly couple the modules. This is the reason why tight coupling corresponds to the case of integral architecture.

By contrast, loose coupling corresponds to modular architecture. In part (ii) of Fig. 3, the interfaces loosely couple modules 1, 2, and 3, which means that function 1 in module 1 does not heavily depend on the functions in modules 2 and 3. This pattern of functional dependencies corresponds to modular architecture (B).

The mapping pattern leads to differences in architectural characteristics. In some cases integral architecture yields better deigns, while, in other cases, modular architecture works more effectively. We will now discuss this point further.

2.5 Worse Is Better: Integral vs. Modular

When we consider the two types of architecture, integral and modular, the obvious question is: which architecture is better? Before trying to answer this very difficult question, let us examine another matter: which architecture is more appropriate with respect to design science?

Modular architecture has weak dependencies among modules and many design options, since design changes are easily implemented. On the other hand, integral architecture has strong dependencies among modules and few design options, since design changes are difficult to make. Therefore, from the point of view of design science, modular architecture is more fitting than integral architecture. However, this does not imply that modular is necessarily better than integral. The classic essay by Gabriel (1989) on software design claims that worse is better, meaning that a design that seems wrong in terms of design science might be proven to be a better design for most of the users. There are two major reasons for this. Firstly, the right design tends to become too complicated. Complete modularity requires a lot of buffers, also called *fat*, which make the design too complicated. Secondly, it is very often the case that the right design causes functionality to be poor because of too much fat.

In other words, the right design is more likely to fail because it lacks balance between design complexity and user needs. Product designers seek to strike this balance, and, although the result might be a scientifically worse design, it will fit the needs of its users.

Figure 4 explains the balance between functions and modules based on the type of architecture. Suppose that the architecture is redundant, like in case (b). Modular system 1 has fat (or buffers) in its modules, introduced by product designers to obtain weak dependencies across modules. In other words, it is the fat that ensures the modularity of the system. If product designers remove the fat, the system loses its modularity and becomes lean, but it still performs the required functions. When a system increasingly loses its buffers, its interfaces couple the modules more and more tightly, and its architecture eventually becomes integral.

Now let us consider another redundant system, shown as (c). The functions in modular system 2 have reduced functionalities. Product designers may opt for reduced functionalities because they do not want a function to be heavily dependent on other functions in other modules. In other words, they want to keep the system modular. Nonetheless, in some cases, they need to increase the functionalities to meet the users' needs and have no choice but to break the modularity of the system. They redesign it so that the function in question heavily depends on other functions in other modules. Through this process, the system architecture becomes integral.

Although modular architecture should be preferable from the point of view of design science, it may not be better in terms of balancing functions and modules. Modular systems are naturally characterized by redundancy, which is necessary for design changeability that makes the system, as a product, more appealing to the

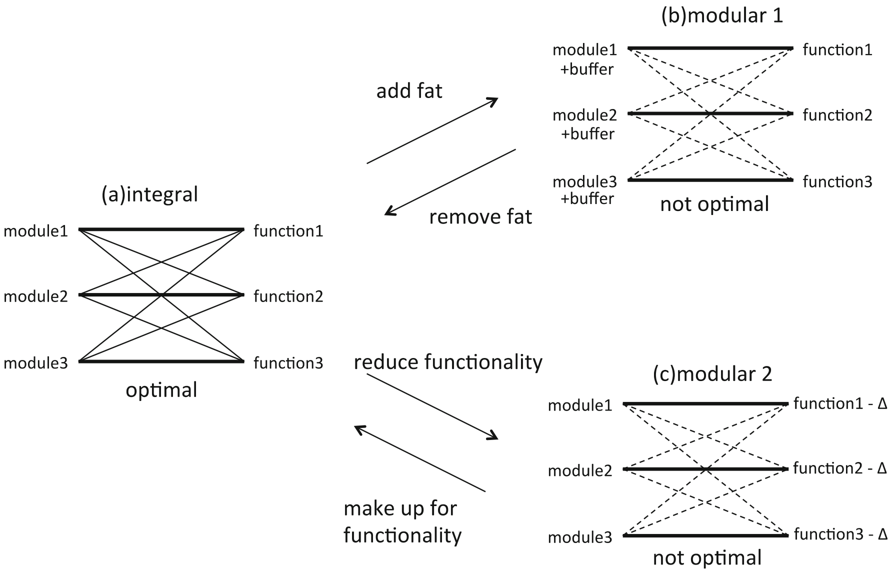

Fig. 4 Worse is better

users, but this upsets the balance between functions and modules. If the system redundancy is removed, its architecture becomes integral. This is the main reason why many successful products are of the integral architecture type. As Gabriel (1989) insisted, product architecture that is appealing to most users is often worse in terms of design science.

2.6 Design Changeability of Modular Architecture

Integral architectures strike an optimal balance under the design dimensions given, which consist of vectors of design parameters. However, integral architecture does not always mean better design, as modular architecture might yield superior results in some situations. Design changeability is the relevant concept here. Figures 5 and 6 show when modular architecture is better than integral architecture.

Figure 5 illustrates the relationship between architecture and optimality. The horizontal axis represents the design dimension, i.e., a set of design parameters for the given modules. The vertical axis indicates the system error, which denotes optimality of functions and modules. System error ε is the difference between maximum system utility, u^*, and expected system utility for a given design parameter d, $u(d)$.

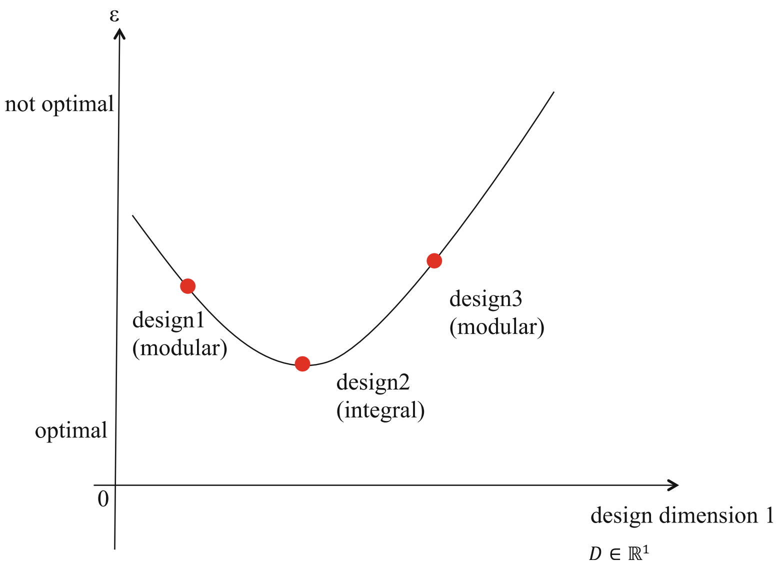

Fig. 5 Architecture and optimality

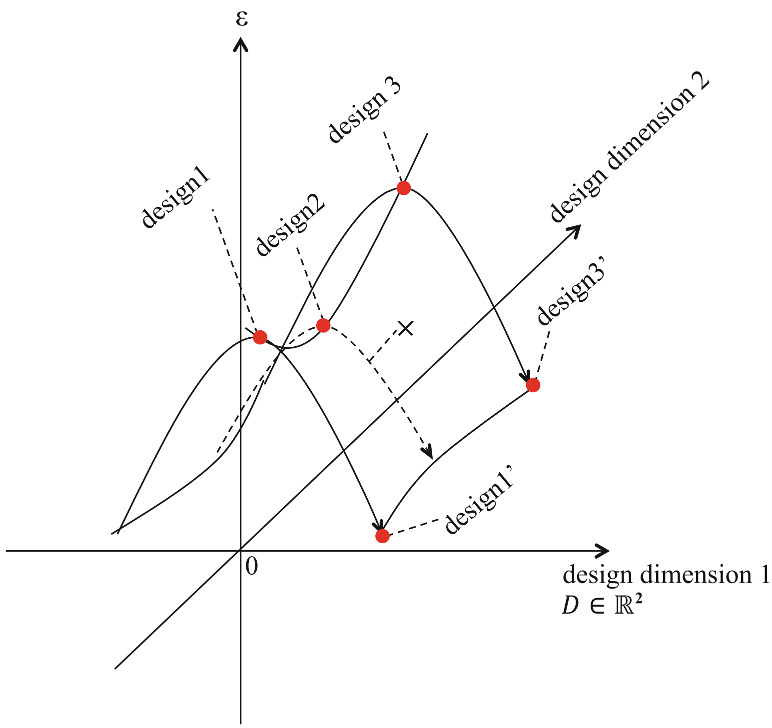

Fig. 6 Expansion of design dimensions

System error ε is defined as:

$$\varepsilon = u^* - u(d) \geq 0$$

$$\begin{aligned} &\text{s.t.} \\ &D \in \mathbf{R}^m \\ &d \in D \\ &u : D \to \mathbf{R} \\ u^* &= \max\{u(d) | d \in D\} \end{aligned}$$

where D is a set of design parameters, m is the number of design dimensions, d is an element of D, u is a utility function, and u^* is maximum utility.

Roughly speaking, expected system utility $u(d)$ means how much utility we expect the system to achieve for a given set of design parameters, d. When the system is optimal, system error ε is zero. Optimality increases moving up along the vertical axis from optimal to not optimal.

In Fig. 5, designs 1 to 3 refer to designs with given sets of design parameters. Each design realizes the system utility and generates the system error. As explained above, the system error is low when the system utility achieved is high. Design 1 is characterized by fat in its modules, which means that some design parameters, packed into modules, do not contribute to realizing the system functions but are simply buffers. In terms of the balance between functions and modules, design 1 is not an efficient design.

On the other hand, design 3 has low functionality. It fully uses the design parameters but still realizes only a small fraction of the needed functions. In other words, design 3 is redundant in its functions. It appears to have many but, in fact, the number of unique functions is small. Just like design 1, design 3 is not an efficient design.

Compared to designs 1 and 3, design 2 achieves greater utility because it has a good balance between functions and modules. It is a lean system under design dimension 1. With respect to architecture, design 2 is an integral system, while designs 1 and 3 are modular systems.

Both designs 1 and 3 have a high system error, which means that a given set of design parameters achieves only low utility, far from optimal. The reason for this is the lack of balance between functions and modules in the system. Too many functions yield a redundant system, while too many modules yield a fat system. Thus, designs 1 and 3 are not optimal, whereas design 2 strikes the optimal balance between functions and modules. In conclusion, integral architecture is the optimal design under design dimension 1.

However, integral architecture has a by-product. When the architecture of a product becomes integral, its interfaces tighten to couple modules, and the addition of other modules becomes problematic. That is, in integral architecture, it is difficult to expand the design dimensions by adding new modules. Conversely, the advantage of modular architecture is that new modules can be added without complications. Modular architecture easily expands the design dimensions by adding modules. If expanding the design dimensions brings major advantages, then modular architecture might be better suited to realizing the design than integral architecture.

Figure 6 shows the relationship between design dimensions and system utility. Under design dimension 1, design 2, which is integral, realizes optimal system utility, while designs 1 and 3, which are modular, are not optimal.

If product designers add another module, the design space expands to design dimensions 1 and 2. Designs 1 and 3 can use the design parameters in design dimension 2. Since their architecture is modular, another module can be easily added. By using the design parameters in dimension 2, designs 1 and 3 can achieve greater utility than design 2. For example, product designers can modify design 3, turning it into design 3′, to improve its optimality. The utility of design 3′ exceeds that of design 2, so that modular architecture makes for better design than integral architecture in this case.

As for design 2, it is not so easy to use the design parameters in dimension 2, since the architecture of this design is integral and its interfaces tightly couple the modules. Thus, adding another module is difficult. If product designers want to use the parameters in dimension 2, they need to loosen the interfaces of design 2, making it more similar to design 1 or 3, and they then can use the design parameters in dimension 2. However, this is a rather tough design process.

In sum, modular architecture is not optimal under the design dimensions given. But, by adding new modules, it easily expands its design space. Under the new dimensions, modular architecture can achieve better designs than integral architecture. With regard to design changeability, modular architecture is superior to integral architecture.

2.7 *Openness of Interfaces*

Another key architectural characteristic is that of the openness of interfaces. The openness or closedness of interfaces refers to the extent to which technological information is shared with other firms. In the case of openness, product designers from one firm can share interface information with other designers from other firms. Conversely, closedness indicates a situation in which product designers can share the interface information only with a few people who have been given permission. These usually range from a limited number of colleagues to selected representatives of the customers or suppliers.

From an industry-wide perspective, information sharing across firms is deeply affected by the degree of standardization of interfaces. Standardization is a coordination process for firms participating in an ecosystem. Open standards, which are the outcome of standardization, represent interface information to be shared for compatibility and can be used by all the firms in the ecosystem. Hence, standardization is an industry-wide activity to enhance information sharing.

Nowadays, most products are complex systems with many interfaces, requiring collaboration among numerous firms. Since the sharing of information is essential

Table 1 Three approaches for open standards

	Setting of standards	Diffusion of standards	Open areas
De facto standards	Market process (dominant firm)	Market process (consumer preference)	Narrow
Consensus standards	Nonmarket process (consortium)	Market process (consumer preference)	Wide
De jure standards	Nonmarket process (committee)	Nonmarket process (often mandatory)	Narrow

for their design tasks, standardization is a key activity for any firm developing complex systems, as it leads to open standards, which are the basis for such collaborative development. The process through which standardization is achieved affects the features of the interfaces of complex systems. Every firm carefully watches the standardization process and its outcomes, since the open standards will influence the growth of the business ecosystem and the competitive advantages gained by firms. Indeed, the main feature deriving from standardization is openness or closedness, which greatly depends on the standardization approaches. The firms that set the standards choose the appropriate approach to standardization according to their business strategies.

The standardization process includes two main phases: setting of standards and diffusion of standards. The setting of standards refers to the designing of interface protocols, while the diffusion of standards is an activity aimed at promoting information sharing about the interfaces. The combination of these two aspects leads to the classification of open standards into three types: de facto standards, de jure standards, and consensus standards. Table 1 provides a summary of three approaches for open standards.

With respect to openness, the three approaches to standardization produce different outcomes. Openness is measured by the size of the open area, i.e., the market segment that does not have entry barriers and is easily accessible to newcomer firms. De facto and de jure standards are likely to produce narrower open areas than consensus standards.

These three approaches are clearly different, especially for what concerns the abovementioned phases of setting and diffusion of standards. However, researchers may confuse them, and, in particular, consensus standards are often mistaken for other standards. This is mainly because consensus standards have appeared more recently and are relatively new to researchers. In addition, consensus standards are partially similar to de facto and de jure standards.

De facto and consensus standards differ in the setting phase. De facto standards are set by a single firm, as there is the risk of infringing antitrust laws if multiple firms gather together to set de facto standards. If they wish to avoid this risk, they should opt for the setting of consensus standards, for which the firms legitimately form a consortium open to third parties. Due to this openness, consensus standards are likely to produce wider open areas than de facto standards.

Turning now to the comparison between de jure and consensus standards, the difference here is in the diffusion phase. De jure standards are ordained by law and are thus often mandatory. Consequently, promoting the diffusion of open standards is not the chief concern in this case. By contrast, consensus standards are voluntary rather than mandatory and are likely to produce much wider open areas for easy diffusion.

There is another reason why consensus standards are characterized by wider open areas. Since they are voluntary, multiple consortia will probably set similar standards and compete in the market. All the consortia care about standards diffusion, which determines the winner of the competition. The wider open area favors the diffusion of standards because it is attractive for newcomers. With the risk of standards wars, consensus standards provide a wider open area through standardization.

In sum, consensus standards usually produce a wider open area than both de facto and de jure standards and are a relatively new approach, whereas de facto and de jure standards represent the old approach to standardization. Consensus standards have been spreading since the mid-1980s through a general relaxation of antitrust laws. This has encouraged the joint development of consensus standards and the creation of wide open areas in business ecosystems. There are many examples of consensus standardization, such as the USB forum for open PC and peripherals interfaces, the AUTOSAR consortium for the car electronics area, W3C for open Internet protocols, as well as Industrial Internet Consortium and OpenFog Consortium for the IoT area.

2.8 Architectural Convergence: Open-Modular and Closed-Integral

So far, we have discussed the key features of interfaces. Firstly, we spoke about tight or loose coupling. A system with tightly coupled interfaces often corresponds to integral architecture, whereas loosely coupled interfaces are likely to go with modular architecture. The second aspect is that of openness or closedness. A system that allows the open sharing of interface information will probably have an open architecture; otherwise, its architecture will be closed.

It can be deduced from the above discussion that system architecture can be described along the two dimensions of integral/modular and open/closed. This yields four possible types of architecture: integral or modular in coupling modules and open or closed in sharing information. Clearly, the two architectural dimensions have interactions because both depend on the features of the interfaces in the system, which reflect the mechanisms of collaborative problem-solving among firms. By virtue of these interactions, the four types of architectures are reduced to two combinations: closed-integral and open-modular architectures.

With respect to joint problem-solving, closed-integral architecture is characterized by a consistent approach. As the interfaces are closed, their information is shared among a limited number of firms. Consequently, each of the modules in the system can connect only with a few other modules. However, the large amount of detailed information shared through intense collaboration helps improve the system and optimize its functions. These design changes sacrifice module compatibility to achieve optimal performance and product integrity by means of tight coupling.

On the other hand, open-modular architecture has a different design approach. Its open interfaces enable product designers to collaborate with several other firms, and, since the latter also include newcomers, new module combinations are easily developed and work as a source of new value for the product. Product designers expect high interface compatibility and can make the most of the open interfaces by experimenting with new combinations.

As far as the design logic is concerned, both closed-integral and open-modular architectures are characterized by consistency. But what about open-integral or closed-modular products? If designers opt for an open-integral architecture, which is an integral architecture with open interfaces, then inconsistency occurs. The interface protocols are often modified to optimize system performance, and this drastically reduces open compatibility, which forms the basis of open interfaces for third parties. Integral architecture does not go well with open interfaces in the design process.

Similarly, a closed-modular architecture, i.e., a modular architecture with closed interfaces, usually proves to be a poor choice. In the modular architecture, product designers can experiment with different combinations to try and create new value in use, but, if closed interfaces are chosen, this prevents them from exploring all the possible module combinations. Hence, they do not make use of modular architecture with closed interfaces because it lacks open compatibility with products from other firms.

As mentioned above, in the closed-integral and open-modular types, architectural convergence comes from consistent design logic. In the closed-integral architecture, top system performance is realized through frequent interface adjustments. Detailed information about the interfaces is shared within a small, tight-knit group of product designers that work on customization for optimal performance. Through this process, the system eventually becomes a closed-integral architecture. By contrast, the open interfaces of open-modular architecture allow product designers to share interface information across firms and try various module combinations to assess new value in use. Modularity is crucial for this trial and error process because it ensures compatibly among modules. Through this process, the system eventually becomes an open-modular architecture. Familiar examples of this architectural convergence are automobiles, a notable case of closed-integral architecture, and personal computers, with typically open-modular architectures. The next section discusses these two architectures in relation to industrial dynamics.

3 Architectural Change and Business Ecosystems

3.1 Digitalization: Trigger of Architectural Change

So far we have discussed business ecosystems and product architecture. A business ecosystem is a form of industrial cluster, and product architecture is the basic design of module configuration. Industrial clusters and product architectures are linked by the design logic according to which firms collaborate to solve design problems. Each type of architecture has its different design logic. Therefore, changes in architecture lead to changes in the industrial clusters.

Two main factors that cannot be ignored in the innovation environment and affect product architectures in most industries are digitalization and globalization. Let us look at digitalization first.

Compared with analog technologies, digital technologies are critically different in the design of system interfaces. Analog technologies are based on the laws of nature and cover mechanics, chemicals, and any other areas relying on physical phenomena. By contrast, digital technologies are based on logical rules, and software and electronics are their main applications. Since they are free from the constraints of physical laws, there is great flexibility in their architectures. This means that product designers can easily decouple one system into multiple modules according to their purposes, such as reducing complexity and redesigning by trial and error. Such decoupling of complex systems is not possible in analog technologies. Moreover, digital technologies are much more versatile than analog ones in the creation of system interfaces and in the organization of their interactions.

Interface design flexibility provides different options for firm collaboration. Closed interfaces are often the result of collaboration within a single firm, while open interfaces are ideal for collaboration across firms. Complex systems require many firms to collaborate, so the interfaces linking their modules must be open, to allow for changes and improvements through trial and error. These changes lead the system to become increasingly open in its architecture. In other words, digitalization spurs a shift toward open architecture.

In addition, digitalization affects the firms' business model. With digital technologies, product designers easily make open interfaces, and third-party firms propose new business models based on the flexible combination of modules. Thanks to the existence of open standards for the interfaces, several firms can supply compatible components. More options for module combination enhance the opportunities for new business models, which often take advantage of network effects. Open interfaces generate network effects, and open standards make for even stronger network effects. Indeed, the platform business is a major example of how to make the most of the network effects of open interfaces and standards.

3.2 Globalization: Amplifier of Architectural Change

Open architecture is a system with many open interfaces whose information is shared by firms to collaborate in design tasks. Since complex systems require the collaboration of product designers from many different firms, they are likely to have open interfaces. Firms also set open standards, such as protocols or references for open architecture, because open interfaces might not be enough for easy interfirm collaboration. Indeed, firms need efficient measures to share information for joint tasks, and open standards are a powerful tool for this purpose. By providing the formal protocols of interfaces, they enable effective cooperation, so that firms can complete their joint tasks. This is why the development of complex systems is often accompanied by open standardization.

As mentioned above, there are three main approaches to open standardization: de facto, de jure, and consensus standards. De facto and de jure standards are traditional approaches, whereas consensus standards have developed more recently. These three approaches ensure flexibility in standardization, and firms choose the most appropriate one according to their strategies.

As well as supporting firm collaboration, open standards also play a key role in industrial clusters, since they remove barriers and stimulate the entry of newcomer firms. This is because open standards consist of explicit information regarding technological knowledge and industrial contexts, so newcomer firms can easily apply it to their products. As explicit information removes the entry barriers that come from implicit knowledge in the industry, open standards stimulate new entries and increase collaboration among firms.

Globalization amplifies the stimulating effect of open standards. Back in the 1980s, firms based in the developed countries were in a dominant position. However, from the 1990s onward, globalization brought about significant changes, as newcomer firms from developing countries entered the global market. Soon, they faced difficulties in joining pre-existing networks of firms because they were not familiar with the technological knowledge and reference industrial context. Yet, open architecture gave them the opportunity to access explicit information, thanks to open interfaces regulated by protocols released to the public by standardization consortia. Thus, open standards removed entry barriers and provided a common base for firm collaboration.

Newcomer firms from developing countries joined the networks of firms with the aid of open standards. Personal computers are a well-known example. In the 1990s Korean and Taiwanese firms entered the personal computer industry, until then dominated by firms from developed nations, such as the US, Japan, and the EU countries. Due to the availability of open standards for PC components, such Korean and Taiwanese firms soon achieved huge success in the global market. Korean firms aggressively invested in the production of memory semiconductors, which had open standards for mutual compatibility, and quickly reached extremely high production capacity. In a relatively short time, they overtook the US and Japanese manufacturers of memory semiconductors and came to satisfy most of the worldwide memory demand. On the other hand, Taiwanese firms successfully developed a new business model known as ODM, or original design manufacturing (Kawakami 2007), which is a special form of contract manufacturing. ODM firms design and manufacture new products based on the specifications of their client firms, such as computer brand firms in the USA. The clients then rebrand the products and sell them as their own. The ODM business model relies on open standards. By using them, Taiwanese firms learnt to flexibly combine various modules and develop new configurations for product design. By the end of the 1990s, Taiwanese firms accounted for around 90 percent of the global production of notebook computers.

Open architecture products fully enjoy the benefits of globalization. The open standards set for their interfaces stimulate the entry of new firms from developing countries and allow international collaboration among firms from developing and developed economies. From a macroscopic perspective, open standards promote the international division of labor.

Product architecture and industrial clusters are linked by the design logic according to which firms collaborate to solve design problems. In today's globalized world, newcomer firms from developing countries need open interfaces and open standards to develop their products in collaboration with other firms. They choose the open architecture model, which often leads to the formation of industrial clusters as business ecosystems, because it generates network effects that allow business ecosystems to grow rapidly and increase their share in the global market. Personal computers are a good example of the linkage between product architecture and industrial cluster affected by globalization. The interplay of architectural and industrial dynamics is analyzed in the next section by comparing automobiles and personal computers.

3.3 Comparison Between Automobiles and Personal Computers

A comparison between automobiles and personal computers is a good way to show how product architecture influences the shape of the resulting industrial cluster. From the point of view of product architecture, automobiles are typically closed-integral, whereas personal computers are open-modular (Fujimoto 2007; Baldwin and Clark 2000). As for the tightness of interfaces, automobiles adopt an integral architecture in which the interfaces tightly couple the modules. By contrast, the interfaces of personal computers loosely couple the modules. Regarding the openness of interfaces, in automobiles the interfaces are mostly closed and accessible only to a limited number of supplier firms, while personal computers have open interfaces, and supplier firms can freely use them.

Table 2 Comparison between automobiles and personal computers

	Product architecture	Closed-integral	Open-modular
	Example	Automobile	Personal computer
Architecture	Characteristics of products	High integrity and optimal performance	Variety of uses with several module combinations
	Features of the interfaces	Interfaces tightly couple modules	Interfaces loosely couple modules
		Small groups of firms share the information	Firms openly share the interface information
	Design logic among firms	Mutual coordination based on relation-specific assets	Autonomous coordination based on open standards
Industrial cluster	Form of industrial cluster	Cohesive clique	Business ecosystem
	Entry barrier	Relation-specific assets work as entry barriers	Open standards remove barriers and allow entry to many newcomers
Recent impact factors	Digitalization	Interface design flexibility gives new opportunities for open collaborations among firms	More open interfaces allow new module combinations for experimenting new value in use
	Globalization	Entry barriers prevent new entries into the global market	Globalization boosts newcomer entries from developing countries
		Through FDI, the network of firms expands globally, but its speed is slow	Business ecosystems quickly grow to global scale

As discussed above, system architecture may be categorized depending on the features of the interfaces and the approaches to coupling modules and sharing information. This corresponds to the two main dimensions of modular vs. integral and open vs. closed, which have interactions that eventually reduce the architectural types to closed-integral and open-modular. These two architectures affect the development of the industrial structures, as illustrated in Table 2, which also reports the effects of digitalization and globalization.

The first column of Table 2 illustrates the closed-integral case. A closed-integral architecture achieves integrity within the system, the interfaces of which tightly couple specific pairs of modules. Intense collaboration among selected firms allows product designers to share all the relevant information, so that they can calibrate the interfaces and attain optimal performance. This design process causes the interfaces to couple the modules more tightly. Due to information sharing at a deep level, firms gain relation-specific capabilities or assets for mutual problem-solving that give them the ability to efficiently develop the products. On the other hand, these relation-specific assets act as deterrent for those wishing to enter the suppliers network, because they are a collection of implicit knowledge and are not imitable by newcomer firms. So, the industrial structure behind closed-integral systems is a form of cohesive clique, characterized by limited membership and deep information sharing.

Automobile is a well-known product with closed-integral architecture. Automakers and suppliers work together on the designing of cars and often change the interfaces between modules to improve performance. Information sharing helps in joint problem-solving, and relation-specific assets allow the existing suppliers to efficiently collaborate with automakers. Yet, third parties do not have access to this information, which represents a deterrent for newcomer firms (Dyer and Nobeoka 2000). Through joint development, the relation-specific assets make the industrial cluster of automobiles a cohesive clique, which is an efficient network of firms with deep knowledge sharing as well as a closed network with barriers to entry.

Turning now to the second column of Table 2, let us examine the open-modular case. In open-modular systems, interface compatibility makes it possible to combine modules in different ways to find the best setup and achieve maximum value in use. The interfaces are open to third parties and firms often set open standards that further reinforce compatibility by providing protocols and references for compatibility confirmation.

Open standards enable supplier firms to access interface information, so that modules are developed in a decentralized manner. By following open protocols, design problems are jointly solved through autonomous coordination, and supplier firms do not depend on relation-specific assets based on deep knowledge sharing and mutual coordination. The coordination mechanism of open-modular architecture, relying on open standards rather than relation-specific assets, significantly differs from that of closed-integral architecture in that it makes it easier for newcomers to enter the market and join the suppliers network. The industrial cluster of open-modular systems consequently becomes a business ecosystem, in which modules with open interfaces are developed through collaboration based on open standards and there are no barriers to the entry of newcomer firms.

If we look at the case of personal computers, we can easily understand why researchers often refer to them as typically open-modular systems. Personal computers have open interfaces based on open standards, and this enables a variety of module combinations that might produce new value in use. In addition, open standards help newcomers enter the PC market. As the number of newcomers increases, module combinations multiply, generating further value in use. Indeed, the entry of newcomers acts as an engine for the growth of business ecosystems, and, as pointed out in previous studies, new firm entries and new module combinations have worked as the main engine for the growth of personal computers (Baldwin and Clark 2000).

The two architectures described here, open-modular and closed-integral, have different principles and mechanisms for collaboration among firms. Open-modular architectures use autonomous coordination based on open standards, while closed-integral architectures use mutual coordination based on deep knowledge sharing. This translates into differences in the shape of the industrial clusters for open-modular and closed-integral products. Through autonomous coordination, open-modular architectures cause industrial clusters to become business ecosystems, while closed-integral architectures, based on mutual coordination, cause industrial cluster to become cohesive cliques. The comparison between automobiles and personal computers clearly shows how different architectures lead to different forms of industrial clusters according to the coordination mechanisms used.

3.4 Interaction Between Digitalization and Globalization

The previous sections illustrated the features of closed-integral and open-modular architectures, which use different approaches to joint problem-solving and yield different forms of industrial structures: cohesive cliques for closed-integral and business ecosystems for open-modular. This helped us shed light on the fact that, when we investigate an industry, we have to take the architecture of its products into account, because the design logic adopted connects its products and the forms of its industrial clusters.

This connection also means that changes in product architecture modify the form of the relevant industrial cluster. Before comparing automobiles and personal computers, we discussed two critical factors altering product architecture: digitalization and globalization. Since the 1990s, their impact has been growing stronger, causing changes in product architectures which, in turn, have impacted on the shape of industrial structures in different ways.

First of all, let us examine the impact of these two factors on the automobile industry. Within the design process, digitalization makes it easier to set the interfaces, because the rules to be followed are only logical and not physical ones, which is particularly useful when the complexity of the system increases. In order to reduce complexity, product designers often decouple the system into multiple modules and adjust the interfaces. They then reconnect the modules in more effective ways to realize the system functions. This design trend is particularly evident in automotive electronics and software, in which the increasing number of interfaces triggers architectural changes toward the open-modular type. As product architecture evolves, so does the form of the industrial cluster, and, with the development of electric vehicles, autonomous driving, and MaaS (mobility as a service), the pressure spurring this change has grown significantly in the developed countries.

On the other hand, globalization affects the automobile industry by increasing the presence of emerging economies, where consumers tend to choose vehicles having the traditional architecture, i.e., closed-integral, so that the market for closed-integral cars is growing in the developing countries. Through foreign direct investment, or FDI, automobile manufacturers from developed countries try to boost their production capacity by collaborating with local automotive manufacturers and suppliers. The resulting industrial cluster is still a cohesive clique. As FDI increases, the cohesive clique grows globally, thanks to new entries, but this industrial change is relatively slow due to the barriers characterizing this business model that act as a deterrent for newcomer firms.

In brief, the cohesive clique of the automobile industry is experiencing a complex transformation, which is a mixture of disruptive change and robust growth. Digitalization mainly drives the disruptive change, while globalization contributes to the robust growth. In developed countries, the pressure linked to this architectural change is on the rise due to digitalization. Nevertheless, vehicle architecture is stable in the emerging countries, whose role is ever more important thanks to globalization. Both digitization and globalization exert pressure on the architecture of automobiles, but in clearly opposite directions, so that automobile manufacturers have to make tough decision to manage this challenging situation.

Now, let us turn to the case of personal computers and how they are affected by digitalization and globalization. As discussed above, PCs can be described as open-modular systems, and the relevant industrial structure is the so-called business ecosystem. Open-modular systems have open interfaces, and, due to digitalization, the number of such interfaces increases, so that open innovation intensifies. Open standards, created by firms to promote collaboration, make the interface knowledge explicit and formal, stimulating the entry of newcomer firms. These newcomer firms rely on these open standards, since they do not have sufficient implicit knowledge of the segment and of its industrial context. The entry of newcomers expands the size of the business ecosystem. Globalization, which enables the creation of a global pool of newcomer firms from developing countries, accelerates the expansion of the business ecosystem at the global level and, in turn, this helps the international division of labor among firms from developed and developing countries.

The case of personal computers shows the connection between architecture and shape of the industrial cluster in open-modular systems. Since the 1990s, digitalization has impacted on personal computers through the setting of open interfaces. Open standards, which guarantee interface compatibility, have generated numerous combinations of modules and realized new value in use while also removing barriers to entry and, consequently, accelerating the growth of the business ecosystem. Among newcomers, memory chip manufacturers in Korea and ODMs in Taiwan have made the most of the open interfaces and standards to earn a prominent place within the global ecosystem of personal computers. Their rapid growth stems from the very features of the business ecosystem, whose open interfaces allow newcomers to create compatible modules and new module combinations for greater value in use. Thus, the personal computer market has become a global ecosystem due to digitalization and globalization.

The consequence of the above has been not only the expansion of the business ecosystem but also the emergence of a new business model, i.e., the so-called platform business, to take advantage of the network effects generated within the ecosystem. Major platform firms, like Intel and Microsoft, have exploited the network effects to the fullest, and their presence has also boosted the business of newcomer firms from emerging economies. Further entries by newcomers have accelerated the expansion of the business ecosystem internationally, thus forming a global ecosystem (Tatsumoto et al. 2009).

3.5 Revisiting Architectural Dynamics

As explained above, digitalization and globalization strongly affect product architecture and the shape of the reference industrial cluster. Digitalization is likely to cause architectural change, and globalization spreads this impact on a global scale. In other words, digitalization is a trigger of architectural change, while globalization is an amplifier of its impact.

For some time, the automobile industry has been showing major signs of architectural change. These trends, referred to as CASE (connected, autonomous, shared, and electric vehicles), are likely to shift the industrial structure of the automobile sector from cohesive clique to business ecosystem. When estimating their impact, it is important to consider where they happen in the architectural hierarchy. The above four trends may be architecturally divided into two categories: changes occurring inside the core layer and changes occurring outside such core layer.

Changes inside the core layer imply architectural changes of the automobile itself and concern innovations such as autonomous driving and electric vehicles. To deal with them, automobile manufacturers and suppliers need to adjust their structures by investing considerable time and effort in organizational reorientation. This is particularly true for large incumbent firms, whose legacy assets are based on traditional architectures. If they fail to keep up with the changes, they run the risk of losing market share as newcomers quickly catch up and gain prominence.

Changes outside the core layer in which automobile manufacturers usually do their business are likely to have an equally large impact and revolve around connectedness and sharing of vehicles. The former feature means that cars are connected to the Internet and have new functions or services based on big data, while the latter means that car users change their behavior from owing to sharing vehicles. These changes affect the upper layers of the whole system, in which the car itself is just one element, and may lead to the creation of new business models that will replace the existing one. For instance, connectedness relies on peripheral services using big data to offer navigation support and customized recommendations. This model profits from displaying ads and matching services. The second aspect, car sharing, might change the landscape of the automotive industry even more drastically. Indeed, if this becomes a mainstream trend, car users will stop purchasing their own vehicles, causing the total disruption of the traditional automobile business model.

Moreover, the above changes may have interactions and become considerably stronger. Indeed, for the time being, they are mainly occurring in developed countries, but, if they spread to the developing countries as well, they will affect the global system. This may come to pass because automobile manufacturers in developing countries have no legacy assets and because car users in those areas might prefer products and services provided by the new business models as they develop their own consumer culture. Hence, electric vehicles and vehicle sharing might prevail in the developing countries.

Although the architecture of automobiles has been stable for years, the CASE trend seems to suggest that the industry is about to undergo a major shift, likely to be accelerated by the emergence of new data technologies, including IoT, big data, and AI. In recent years, these data technologies have had an impact on all the industries and have become a common technological framework (The Economist 2017). They use data resources coming from the products and realize new value in use, thus intensifying the architectural changes in the automobile industry.

Product architecture is dynamic, and, although it appears stable, dramatic changes can occur at any time, as confirmed by the evolution of the personal computer industry. The architectural change from mainframes to personal computers utterly transformed the landscape of the computer business. Although automobile manufacturers have enjoyed the benefits of a stable architecture for years, recent signs of architectural change are forcing them, especially incumbent firms, to find new approaches.

Nonetheless, we are still in the early stages of this shift, and for now it is impossible to determine with any degree of certainty whether the automobile industry has reached an inflection point. Digitalization has accelerated the above shift and might bring about new business models, but so far globalization has supported the stable growth of the existing architecture. Yet, car users in developing countries might decide to choose products and services that disrupt the traditional business model of the automobile industry, and automobile firms need to learn to cope with this difficult situation.

4 Conclusion and Further Insights

This chapter illustrates the concept of business ecosystem as a special form of industrial cluster and explains how the design logic connects product architecture and the shape of the industrial cluster. It then discusses architectural changes and their impact on the shape of industrial clusters by drawing a comparison between automobiles and personal computers.

Industrial clusters are likely to become cohesive cliques when the product architecture is closed-integral. Conversely, business ecosystems are likely to emerge when the product architecture is open-modular. Business ecosystems have a special growth model because they contain many network effects, and firms seek new business models to exploit them. The platform business is a notable model among them that has exploited network effects as a source of new value in use to grow and reach a global scale.

Industrial clusters are shaped by the design logic adopted to develop the architectures of their products, which relies on different types of collaboration among firms. So, architectural changes affect the form of an industrial cluster, and, if its product architecture changes from closed-integral to open-modular, the cluster will shift from cohesive clique to business ecosystem.

Product architecture can be modified by various factors, and most industries have recently been affected by two phenomena that they cannot ignore: digitalization and globalization. Thanks to digitalization, product designers can now flexibly set interfaces with open standards in order to reduce product complexity. These open interfaces can shift the product architecture toward open-modular. Globalization, characterized by the strong presence of emerging economies, causes the architectural changes to reach global scales. Digitalization works as a trigger for architectural change and globalization acts as its amplifier.

The dynamics of product architecture are constantly evolving, and this is why architecture can drastically change even when it seems stable. Due to digitalization, the automobile industry is now showing signs of major architectural shifts. The pressure is stronger than ever before. Firms need to focus on macro aspects because changes are taking place both inside and outside the core layer where automobile manufacturers mainly do their businesses. Changes inside the core layer, such as autonomous driving or electric vehicles, bring about architectural changes that might transform the shape of this industrial cluster. On the other hand, changes outside the core layer, such as connected cars or vehicle sharing, create new business models that disrupt the traditional practices of the automobile industry. Globalization might amplify the impact of these architectural changes on the global market, since firms from developing countries, which are newcomers in the global economy and have no legacy assets, may choose to adopt new architectures. In addition, car users in developing countries might prefer product or services based on these new architectures closer to their own consumer cultures. To deal with these architectural dynamics, it is crucial to understand the connections between product architecture and industrial cluster.

Acknowledgment This work was supported by KAKEN Grant Number JP16K03850.

References

Aoki M (2001) Towards a comparative institutional analysis. MIT Press, Cambridge, MA

Baldwin CY, Clark KB (2000) Design rules: the power of modularity. MIT Press, Cambridge, MA

Dyer JH, Nobeoka K (2000) Creating and managing a high-performance knowledge-sharing network: the Toyota case. Strateg Manag J 21:345–367

Eisenmann T et al (2011) Platform envelopment. Strateg Manag J 32:1270–1285

Evans DS et al (2006) Invisible engines. MIT Press, Cambridge, MA

Fujimoto T (2007) Architecture-based comparative advantage: a design information view of manufacturing. Evol Inst Econ Rev 4:55–112

Gabriel R (1989) The rise of “Worse is Better”. https://www.jwz.org/doc/worse-is-better.html. Accessed 9 Apr 2018

Gawer A, Cusumano MA (2002) Platform leadership: how Intel, Microsoft, and Cisco drive industry innovation. Harvard Business School Press, Boston

Grove AS (1996) Only the paranoid survive. Doubleday Business, New York

Iansiti M, Levien R (2004a) The keystone advantage: what the new business ecosystems mean for strategy, innovation, and sustainability. Harvard Business School Press, Boston

Iansiti M, Levien R (2004b) Strategy as ecology. Harv Bus Rev 82. https://doi.org/10.1108/eb025570

Kawakami M (2007) Competing for complementarity: growth of Taiwanese notebook PC manufacturers as ODM suppliers. In: Sato Y, Kawakami M (eds) Competition and cooperation among Asian enterprises in China. Institute of Developing Economies, Chiba. http://www.ide.go.jp/library/English/Publish/Download/Report/2006/pdf/2006_04_30_04.pdf

Nalebuff B (2004) Bundling as an entry barrier. Q J Econ 119(1):159–187

Paine RT (1966) Food web complexity and species diversity. Am Nat 100:65–75

Rochet J, Tirole J (2003) Platform competition in two-sided markets. J Eur Econ Assoc 1 (4):990–1029

Simon HA (1969) The sciences of the artificial, 1st edn. MIT Press, Cambridge, MA

Tatsumoto H (2017) Platform kigyou no global senryaku, Yuhikaku, Tokyo (in Japanese)

Tatsumoto H et al (2009) Platforms and the international division of labor: a case study on Intel's platform business in the PC industry. In: Gawer A (ed) Platforms, markets and innovation. Edward Elgar, Northampton, pp 345–369

The Economist (2017) Data is giving rise to a new economy. The Economist, Fuel of the Future May 6, 2017. Available at https://www.economist.com/news/briefing/21721634-how-it-shaping-updata-giving-rise-new-economy

Ulrich KT (1995) Product architecture in the manufacturing firm. Res Policy 24:419–440