Applying Modular Operators in Platform Development: Case of NTT DoCoMo i-Mode

Jerome Chih-Lung Chou\textsuperscript{a*}, Chia-Liang Hung\textsuperscript{b}

\textsuperscript{a}Faculty of Information Management, Hwa-Hsia Institute of Technology, Taiwan
\textsuperscript{b}Faculty of Information Management, National Chi-Nan University, Taiwan

Accepted 4 June 2008

Abstract

Modular operators can be regarded as maneuvers of platform development. In the context where system interfaces can be standardized and system components can be developed and marketed by a group of technologically and commercially interdependent firms, direct and indirect network effects are crucial to a platform’s success. In this article, we synthesize theories of modularity and network economics to provide a theoretical framework of platform development. The framework shows the applicability of modular operators in developing a platform, and helps the platform provider leverage the force of network effects by platform strategy-making.

Keywords: Modularity, network effect, platform strategy

1. Introduction

Platform strategy is a product strategy whereby the firm uses common modules as the base to develop a stream of derivative products to efficiently target multiple market segments (McGrath, 2001; Meyer and Lehnerd, 1997; Meyer and Seliger, 1998; Meyer and Mugge, 2001). The common module setting design rule is called a platform, and is obtained from system modularization. Controlling an industry-wide platform is driving and channeling the industry’s innovations, and more and more firms want to become platform leaders (Gawer and Cusumano, 2002). Baldwin and Clark (2000) proposed six modular operators to characterize basic patterns of system modularization. Modular operators seem to be a useful conceptual tool for the firm to formulate platform strategies. The purpose of this article is to develop a theoretical framework of a successful platform development based on actions of modularization considering network effects. We reviewed literature of modularity and adopted a value-creating perspective to view system modularity in a value network (Christensen and Rosenbloom, 1995) wherein connections between modules are of both technological and transactional in nature. More than only enhancing product design efficiency, the platform must be designed to facilitate desirable transactions and thereby to incorporate the force of network effects into the strategy. By applying positive feedback loops of direct and indirect network effects as a framework, we demonstrate that actions of modularization and demodularization can be arranged to help the firm seek sustainable industry-wide platform leadership. Lastly, we analyze the development of NTT DoCoMo i-mode to demonstrate the applicability of the theoretical framework.

\* Corresponding author. E-mail: jerome@cc.hwh.edu.tw
2. Modularity and the platform

2.1 Objectives of modularity

Modularity is a general system concept. It is a continuum indicating the degree to which a system’s components can be taken apart and recombined efficiently. A high degree of modularity refers to both the loose coupling among components and more freedom of mixing and matching of components for which the system architecture allows.

Modularity is indeed a solution for humans to conquer the complex design tasks as was mentioned by one of the Nobel Prize winners, Herbert Simon, who put forward early theories about how to design complex systems. When the complexity of some part of the system escalates, defining a separate abstraction that has a simple interface will isolate that complexity and be eligible to be designed as a module for designers to tackle the difficulties of fixing and configuring. While tackling the complexity, the designer usually designs a module as a subsystem for performing a reduced set of functions in the system. Ulrich (1995) defined modular architecture as a one-to-one mapping of functions and physical elements, and the interface must be decoupled. The objective of modularization can be extended to meet commercial requirements, such as fulfilling the needs of certain market segments, reducing production costs, or more. Sanchez and Collins (2001) defined modularity with more commercial considerations. They defined it by saying that a system is modular when the interfaces between functional elements allow variations of components to be substituted into the architecture, and interfaces are not allowed to change during the “commercial lifetime” of the product.

Sanderson and Uzumeri (1995) suggested that the commercial success of Sony Walkman was because it could more easily provide component-based functions, features, and performance levels to saturate the market. Meyer and Lehnerd (1997) made similar arguments by analyzing the case of the Black and Decker power tool. Earlier, Garud and Kumaraswamy (1995) researched the cost-side benefit of modularity. They proposed that if the organizational design can support the modular product design, benefits of substitution could be derived from the reduced performance slippage, amortization of initial design cost over generations of products, and reduced incorporation and searching costs for technology.

2.2 Platform design strategies

In these studies, modularity is a way to enhance a firm’s productivity in design. To reap the benefits of modularity in design, the firm must plan its product platforms. The conventional wisdom is that with fine product platforms, the firm can introduce products covering multiple market segments more speedily and economically. Moreover, with the ability of introducing new products, a firm can also learn the market needs more speedily and economically to achieve a competitive advantage. However, nowadays many system products are not designed by a single firm. Because of the reduced coordination cost due to modularity enabled by interface standards and the efficient communications enabled by the Internet, a constellation of individuals, firms, alliances, and consortia could collaboratively develop system products with decentralized decisions, but on the same rhythm. To reap the benefits of modularity design in such a context, the strategic logic of product platforms seems insufficient, because the technological decisions for developing derive products are not necessarily made by the platform module providers centrally. The economies of scope due to reuse of platform modules, the benefit of learning and the benefit of flexibility are no longer easily realized in such contexts.

In this article, we use the term module to refer to a subsystem designed by a specific organization. On a system diagram, a module is a group of design tasks that are densely interrelated within the group that could be done by one organization with only loose
connections to the other parts of design tasks. The connections between modules link organizations not only in technological aspects but also by transactions. As the number and complexity of such connections decrease, the degree of modularity of the system increases. Sometimes, an organization may provide multiple modules for a system product especially when the interrelated design tasks call for a strategic hierarchical control to coordinate them.

3. The manipulation of modularity: modularization and demodularization

3.1 Benefits of increased modularity

Baldwin and Clark (2000) have identified six modular operators that characterize modularization. These operators are splitting, substituting, augmenting, excluding, inverting, and porting. Definitions of these operators are given in Table 1.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Splitting</td>
<td>Separating systems into components which interact across defined interfaces</td>
<td>Interchangeable drives, keyboards, mice, monitors, and printers</td>
</tr>
<tr>
<td>Substituting</td>
<td>Switching between components which perform the same function</td>
<td>Replacing a Pentium CPU with a Centrino CPU</td>
</tr>
<tr>
<td>Augmenting</td>
<td>Adding a module to increase the functions of a system</td>
<td>Attaching a Web camera</td>
</tr>
<tr>
<td>Excluding</td>
<td>Removing a module to reduce the functions the system can perform</td>
<td>Removing a floppy disk drive</td>
</tr>
<tr>
<td>Inverting</td>
<td>Making an imbedded function into a stand alone module and setting the module’s interfaces</td>
<td>Separating the operating system from DEC’s system to create Unix</td>
</tr>
<tr>
<td>Porting</td>
<td>Moving a module from one system to another</td>
<td>Using a Mac printer on a PC network by adding a translator</td>
</tr>
</tbody>
</table>

Source: Summarized and revised from Baldwin and Clark (2000)

By definition, modularization partitions erstwhile interconnected networks of tasks into discrete sub-networks, called modules, which can still function together because they jointly recognize and follow a set of design rules. The settlement of module interfacing rules for various applications becomes the platform.

Modular operators represent patterns of modularization processes. Among them, splitting and inverting will create new interfaces. Substituting does not result in new interfaces, but improves the system’s local performance in a predictable way. Augmenting and excluding are paired operators to add and subtract modules on the defined interfaces to search for ways to create value. The porting operator links two previously separate systems by a translator, redistributing the add-on value of modules. Especially when the ported module is also a platform, new opportunities may emerge by interactions between the two platforms.

3.2 Limits of modularity

However, the degree of modularity need not increase forever. If the product performance achieved by mix-and-match is satisfactory, integrators will aggregate the platform with some
other modules to produce complete system products without changing the interface. That is, integrators are keeping the degree of system modularity, but increase the variety of derived system products. In contrast, if the performance is unsatisfactory due to the technological constraints of the interface standard, component providers or integrators may try to integrate some modules by existing interfaces, and somehow lower the degree of modularity, to explore the technological and commercial potential of new system products or new components. New platforms may emerge in this way. Some inverting actions integrate with previously separate modules to provide more functionality or higher performance. Schilling (2000) argued that the inter-firm product modularity decreases when component specialization achieves an insufficient functionality of a system’s product, or when customers face difficulties in assessing the quality and in assembling the components. In other words, by knowing the conditions of the network, the platform module provider may integrate strategically the platform’s functions to be less dependent on outside components as well as to purposefully make the system less modular. In this way, inverting can also be regarded as an operator of demodularization.

The successful system product evolves usually through the module providers’ manipulation of modularization and demodularization actions, which are guided by competitive motives in order to construct a favorable value network around the module, and to seek platform leadership in the modular cluster. That requires a strategy.

4. Applying modular operators in platform development

The conventional wisdom of continuously renewing and cannibalizing the platform or the architecture (Morris and Ferguson, 1993; Meyer and Lehnerd, 1997; McGrath, 2001) needs to be broadened. Gawer and Cusumano (2002) pointed that the decisions of product technology are interweaved with decisions of the scope of the firm, external relationships, and internal organization. When making modularization decisions, the firm shall consider issues beyond product technology, and as shown in Table 1, the firm does have alternatives other than just substituting the platform module. Besides, the firm can strategically integrate beyond the firms’ boundary, aggregate complements, and provide self-assisting interfaces for value configuration.

How to enlarge the platform base? Guided by network effects, firstly, the firm should consider inverting and porting because these two actions determine the platform’s compatibility and hence the market size and competitive conditions. The platform provider can continuously substitute the platform in order to make conditions for direct network effect to rise, to improve the platform performance, to keep controlling the interfaces in order to defend away rival platforms, or to create more innovative opportunities for complementors. The platform provider can apply porting to interlink with multiple platforms to bring more opportunities, because the redefined system scope enables opportunities from multiple platforms to interact. Provision of a fine translator module, that is, porting, will reduce complementary design costs, and thereby increase the value of opportunities to complementors.

Secondly, the platform provider should extend the design opportunities, so as to diversify the complements. Beyond taking opportunities due to the growth of installed base, the complementor or platform provider can still employ augmenting, substituting, and porting to make various complementary components in the following life cycles. Augmenting adds components of new functions; substituting adds components with the same function but with various levels of performance; porting adds components from other systems erstwhile. Hence, based on a large variety of complementary modules, the system integrator, or the end-user can assemble, integrate, and configure more varieties of derived systems to search and meet
customer needs. By manipulating the modular operators, they can select new components into the system product. However, the platform providers must carefully govern the conflicts among complements and perfectly adjust the platform core.

Thirdly, to pursue direct network effect, the platform module provider can enhance the system value by inverting or augmenting components with advanced functions into the platform. Such functions must enable end-users to exchange information mutually, or to add value on specific formats, so as to achieve direct network effect and the consequential lock-in result.

We advocate that if there is an opportunity to make the system products or services with demand-side economies of scale, the platform provider should take advantage of this opportunity by altering the platform’s function to approximate a whole product. In so doing, the platform provider can internalize the positive network externalities. If such an opportunity is not obvious or does not exist, platform upgrades can be planned to pull in resources for complementary innovations to search for and expand customer needs, and on the other hand it may reduce the total cost of development of the system product.

While pursuing indirect network effect, in order to channel the direction of complementary innovations, the platform provider can elicit the complementors’ substituting, poring, and augmenting actions, and elicit the integrators’ assembling, integrating, excluding, and configuring actions by providing exemplifying actions. On the other hand, in order to signal to complementors that their innovation opportunities are ensured, the platform provider may deliberately not augment certain complementary modules to encourage a particular complementor. To keep complementors from threatening the platform status, the platform provider should foster multiple complementors and confine their value contributions to be relatively small. Hence, complementors in large amounts should be fostered and supported well to sustain the platform’s central role. As the variety of complementary designs soar, the platform provider should provide tools for integrators to overcome transaction difficulties and fully take advantage of such component variety to roll out various products.

In addition to enlarging the installed base, the platform module provider can interlink the platform with other platforms (that is, porting the platform onto other platforms) to let opportunities from multiple platforms interact and increase the value of spillover opportunities. In summary, modular operators can be properly applied to help start and reinforce positive-feedback cycles to trigger direct or indirect network effects in the network context. The system dynamic cycles are summarized in Figure 1.

**Figure 1. Applications of modular operators in platform development**
5. Case of NTT DoCoMo i-mode

NTT DoCoMo i-mode is known as a successful platform of wireless information service in Japan. Introduced in February, 1999, it achieved one million subscribers in 6 months, and ten million in 18 months, presenting a typical explosional growth rate. I-mode is famous for its content variety and attractiveness, but what is less known is that in developing the NTT DoCoMo i-mode, the e-mail service was the planned key function. Although restricted in features, e-mail service was mainly provided through handsets in Japan. E-mail was a popular function of i-mode, and it was reported that 42% of users used i-mode phones primarily for sending e-mail; 34% used it mainly for making calls (Farhoomand, 2003). About 50% of the usage of i-mode is from e-mail. The usage pattern of e-mail created direct network effect among i-mode users. Based on e-mail and improved bandwidth, now i-mode users can exchange pictures (the service is called i-shot) and video clips (called i-motion).

In pursuing indirect network effect, to expand the content offerings or the installed base first was a chicken-and-egg dilemma for NTT DoCoMo. The company broke this vicious circle by determining which technological standards would be suitable to elicit the participation of content providers. NTT DoCoMo selected C-HTML as the programming language for content editing. The language is a compacted version of the popular HTML and designed under restrictions of the wireless infrastructure, such as the limited bandwidth, small screens, and limited functionality of the devices. After removing certain features of conventional HTML, such as tables and frames, the speed of content delivery could be substantially increased. By selecting C-HTML, NTT DoCoMo significantly lowered the cost of porting the existing wired Internet content onto the i-mode platform. With a generous profit-sharing policy, a wealth of content offerings was generated. These content providers retained porting, augmenting, and substituting contents and services on i-mode.

In managing a variety of contents and services, predicting what the subscribers will consider cool or fashionable is difficult. Fortunately, NTT DoCoMo smartly coped with this problem by letting the market decide the winners. NTT DoCoMo encouraged competition among allied vendors by selecting a few competing contents into the same category under the portal menu, and allowed several independent sites providing similar content to be accessible, too. In so doing, NTT DoCoMo was able to introduce a huge variety of content, much of which probably seemed redundant; however, given the difficulty of predicting which content would succeed, such variety and redundancy were the means of handling the risk. This strategy tends to encourage the production of fresh and trendy content while weeding out the failed ones. Besides, the users could customize their own favorite mobile information by easy tools. As a result, the add-on value and lock-in effect on users increased, keeping the company ahead in the mobile service market.

Moreover, the company has actively coordinated the upgrading of system architecture, and has closely collaborated with handset manufacturers to evolve the platform to support more variety of complementary services even by using a compensational inducement for novelty risk. Whenever new technologies such as Java, color display, embedded camera, and embedded-contactless IC has enabled new services, NTT DoCoMo has rolled out new handsets and thereby the value proposition of i-mode is renewed. NTT DoCoMo has broadened the i-mode data service from content to interactive services, ticketing, ID entry services, and eMoney.
How is NTT DoCoMo applied modular operators in developing the i-mode platform is summarized in Table 2. This case shows how the company successfully applied modular operators, together with integrating and configuring, to support its i-mode platform strategy. The positive-feedback loops achieved by i-mode platform strategy and business strategy are described in Figure 2.

Table 2. Applying modular operators by NTT DoCoMo i-mode

<table>
<thead>
<tr>
<th>Operator</th>
<th>NTT DoCoMo’s actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Splitting</td>
<td>Decoupling the i-mode service platform and the underlying network</td>
</tr>
<tr>
<td></td>
<td>Decoupling the i-mode platform and content/service applications</td>
</tr>
<tr>
<td>Substituting</td>
<td>Rolling out a series of i-mode handsets</td>
</tr>
<tr>
<td>Augmenting</td>
<td>Only adding new supporting functions to the platform</td>
</tr>
<tr>
<td></td>
<td>Not competing in providing applications</td>
</tr>
<tr>
<td>Excluding</td>
<td>Competition for being listed on the i-mode portal site can weed out weak content/service</td>
</tr>
<tr>
<td>Inverting</td>
<td>C-HTML, Java, and Felica interfacing specifications become design rules of complementary innovations</td>
</tr>
<tr>
<td>Porting</td>
<td>Adopting C-HTML to link with Internet</td>
</tr>
<tr>
<td></td>
<td>Embedding Sony Felica contactless ICs in handsets to link with Felica network</td>
</tr>
<tr>
<td>Integrating*</td>
<td>Closely collaborating with manufacturers in designing handsets</td>
</tr>
<tr>
<td></td>
<td>Making its portal site in the form of a starting menu on the handset</td>
</tr>
<tr>
<td>Configuring*</td>
<td>Letting users be able to bookmark their favorite sites</td>
</tr>
</tbody>
</table>

Note: *Operators not mentioned in Baldwin and Clark (2000)
6. Discussion

In addition to maneuvering modular operators, NTT DoCoMo’s near monopoly position in the wireless market and the industrial influence had catalyzed i-mode’s success. As Steinbock (2003) noted, even after the deregulation of the Japanese telecommunication market, NTT DoCoMo still occupied near 60% of market share, or of over 40 millions users. The large market share and industrial leadership made it easier to coordinate the development of a whole service system, especially by integration. NTT DoCoMo has been able to specify the handset features to meet new service requirements with the confidence that the handset makers will comply because NTT DoCoMo had secured a big amount of handset orders from them (Funk, 2001). Moreover, NEC, Matsushita, and other key i-mode handset makers have brought in the electronic expertise to enhance i-mode technology due to the specific learning and co-evolutionary experience with NTT, the parent of NTT DoCoMo. Their relationship has been built well long before since the past collaboration in Japan MITI’s National Computer Program, and can be evidenced by observing NTT’s well-running supply chains. This conventional cooperative relationship increased the possibility for integrating improvements or innovations within the system.
On the content provider side, the large market share and the economy of scale indeed attracted the content providers’ interests. The managed competition mechanism implemented on the i-mode portal menu also helped NTT DoCoMo learn the market demand efficiently. NTT DoCoMo sets categories, and content providers compete by popularity to be listed on some category of the portal menu. Those popular sites will become i-mode official sites and be advertised by NTT DoCoMo, and thus be more easily accessed by users. Accordingly, in addition to easy self-configuring feature of i-mode, NTT DoCoMo successfully maneuvered the configuring operator for selecting the fittest services through the market competition mechanism.

On the user side, prior to the introduction of i-mode, PC popularity and thus the wired Internet experience of Japanese society were low because of some cultural, economic, and regulatory reasons. Among them is the fixed-line giant NTT continued insisting on high charges for local calls and rejecting opening its lines for cheaper Internet services. This resulted in the slow spread of PC-based Internet, and for many Japanese, their first Internet experience was on i-mode. In other countries, because people have more PC-based experience, the variety of wireless content on i-mode seemed less attractive and less effective in promotion (Hung and Yeh, 2007).

These specific factors, also illustrated in Figure 2, are just catalysts to leverage the modular operators and contribute to i-mode’s success in Japan. Nevertheless, NTT DoCoMo i-mode gave a good example of maneuvering modular operators in both technology and business platform development.

7. Conclusions

Modularity permits system developers to hide the complexity of technology inside modules. Modules enable developers to devide innovation tasks, and facilitate transactions of technologies encapsulated in modules among developers. The modular compatibility supports the formation of transactional relations among system developers and end-users that weave into a value network. In the value network, bigger successful developers are those who can make their modules common cores needed by many other developers, and those system products based on their core modules are needed by many end-users. In other words, bigger successful developers are positioned at the center of value networks, and they provide platforms.

In the modern networking context, the decisions of modularization and demodularization should be made according to the contextual conditions. These conditions can be lined along two causal paths, namely, the positive-feedback cycles of direct and indirect network effects. Platform strategy in the network context can be understood as applying modular operators in conjunction with integrating and configuring, and to adjust these conditions to complete positive-feedback cycles to strengthen the platform provider’s ability of creating value and appropriating rent.

Complementary modules with demand-side economies of scale should be augmented or inverted, and integrated into the platform module. Because such modules may result in high market demand and switching cost, and once the collective switching cost gets sufficiently high, the complementary module provider may gain power over the platform provider and may become a platform competitor. The platform provider should de-modularize the system by integrating modules with such potential. While pursuing indirect network effect, the platform provider should select complementors from those who contribute to the adoption of system products only through the increase of component variety.

The case analysis of NTT DoCoMo i-mode provides an evidence that product designers and strategy makers can effectively apply modular operators to incorporate the force of
network effects in platform strategy and hence to support business strategy in the network context.

References