PLATFORM STRATEGY IN NETWORK CONTEXT: MICROSOFT .NET CASE

Jerome Chih-Lung Chou, Department of Information Management, Hwa-Hsia Institute of Technology, Taiwan, R.O.C.  
jerome@cc.hwh.edu.tw, +886-2-8941-5143 #119

Chia-Liang Hung, Department of Information Management, National Chi Nan University, Puli, NanTou County, Taiwan, R.O.C.  
clhung@ncnu.edu.tw, +886-49-2910960 #4640

Nai-Chang Cheng, Graduate Institute of Global Business and Strategy, College of Management, National Taiwan  
chengnaichang@gmail.com, +886-919873513

Tse-Ping Dong*, Graduate Institute of Global Business and Strategy, College of Management, National Taiwan Normal University, 162, He-ping East Road, Section 1, Taipei 10610, Taiwan, R.O.C.  
Corresponding: tpdong@ntnu.edu.tw, +886-2-7734-3310

ABSTRACT

The logic of platform strategy that contributed to the success of walkman or power tool needs to be reexamined when the product is modularized, interfaces are standardized, and derived products are developed and marketed by a group of technologically and commercially interdependent firms. In such context, network effect determines the success of industry-wide platform strategy. In this article, the authors use direct and indirect network effects as an analytical framework to examine Microsoft .NET strategy and conclude that modules with potential of direct network effect are good candidates for the platform leader to integrate into the platform function.

Keywords: system modularity, network effect, platform strategy

INTRODUCTION

Platform strategy is a product strategy whereby the firm uses a common module as the base to develop a stream of derivative products to target multiple market segments efficiently (Meyer & Lehnerd, 1997; Meyer & Seliger, 1998; McGrath, 2001). The common module is called platform and is obtained from system modularization. Controlling an industry-wide platform is taking the position of driving and channeling the industry’s innovations, and more and more firms want to become a platform leader (Gawer & Cusumano, 2002). Baldwin and Clark (2000) proposed six modular operators to characterize basic patterns of system modularization. Modular operators also seem to be a useful conceptual tool for the firm to formulate strategies for developing architectural module, that is, platform. It is observed that companies that are more successful provide industry-wide platforms like Microsoft’s operating systems and Intel’s
CPU chips. The purpose of this study is to examine the development of Microsoft .NET, one of the leading platforms in network context, to find new strategic implementations for platform development.

**SYSTEM MODULARITY IN NETWORK CONTEXT**

Recent cases in IT industry have shown that companies providing modules instead of providing complete system products are usually more successful. These more successful companies adopted a modular system structure and a common standard of compatibility rather than an integral (or modular but prepackaged closed) structure. The companies relied heavily on an external network of complementors and integrators to create value.

System modularity is a technological decision, but to create value with technology, companies also need to design transactions to relate their modules to the economic system. They must transact their modules with other module providers, integrators, or end-users to realize the value. High system modularity allows companies more latitude to devise transactions to support the formation of a value network around the module. Value network is a nested commercial system comprised of module providers, integrators, end-users and markets through which tradable modules are designed, made, and sold to integrators at the next higher level of the subsystem. On the value network, ultimately the end-user defines an architecture of system-of-use in which the end-product is also a module relating to other modules. This definition of value network is adapted from that of Christensen and Rosenbloom (1995). Deciding the product’s modularity becomes a lever for the firm to shape its external relations in the network context.

A proper level of modularity can help grow the value network and create value. System modularity enables the firm to open its technology for attracting complementary or competing module providers and letting end-users have alternatives. Low level of system modularity limits the firm to apply an open strategy to grow the value network but helps to control the product’s intellectual property. The degree of modularity is a compromised decision between value network growing and technology controlling.

Fleming and Sorenson (2003) argued that modular designs make R&D more predictable, but they tend to result in incremental product improvements instead of important advances. On the other hand, coupled designs are riskier to work with, but they are more likely to lead to breakthroughs. This phenomenon can be understood as that in the network context, because network participants are enjoying the benefit due to network externalities, for an incompatible new technology, only significantly beneficial advance that can compensate the loss of benefit will be considered by the modular cluster. In the modular cluster, because of the benefit of coordination by existing interface standards and previous investment of reusable modules, small local performance improvements by modular innovations (Henderson and Clark, 1990) are more frequent than significant architectural breakthroughs. While deciding the degree of modularity, the firm must also trade off the technology predictability with potential innovation breakthroughs.

In summary, the firm must trade off between technology diffusion and control, and between technology predictability and breakthrough possibility while determining the degree of modularity of its product. The degree of modularity will influence the firm’s ability of utilizing outside resources, and the firm must transform the outside resources into a value network. The
firm’s modularization, demodularization, and compatibility decisions affect its external relations and thereby the positioning of the firm on the value network. For leading technology firms that have unflung products with market uncertainty, modular architecture allows such firms to leverage their advantage in technology to drive other suppliers to explore the market or to enlarge the market.

While these decisions are made in the network context, economics of the network need to be concerned. Katz and Shapiro (1994) classified network effects into direct network effect and indirect network effect. Direct network effect happens in communication networks where various end-users join a system that allows them to exchange messages with one another. The value of membership to one user is positively affected when another user joins and enlarges the network. Indirect network effect arises in the common case in which users make their purchases over time, either because end-users enter the market at different times, or because the end-user may spread purchases over time. In such markets, adoption externalities come about indirectly, through the impact of one end-user’s adoption decision on the future variety or prices of components. The durable module that enables integrators to derive various system products is called platform. The platform provider is embedded at the center of a value network that consists of interdependent firms and end-users. The mechanism of direct and indirect network effects is depicted in Figure 1, and is served as an analytical framework to examine the development of Microsoft .NET platform. The platform provider’s endeavors of changing system modularity and facilitating transactions with other network participants are identified and analyzed guided by the framework.

![Diagram](image)

**FIGURE 1 Network Effect**

**CASE ANALYSIS: MICROSOFT’S .NET STRATEGY**

Microsoft .NET embodies the company’s vision of “software as services.” .NET is a development environment for Web Services that is considered as a new generation of Internet technology. Web Services are reusable software components capable of being shared across the Internet. Microsoft is currently in the process of building the service grid and rewriting all of its major products from Visual Studio to Office to incorporate Web Services standards.

There are three positive-feedback cycles in .NET strategy need to be managed. First is the
positive-feedback cycle of expectation resulted from Microsoft’s involvement and leadership in WS-I organization. Microsoft and IBM jointly initiated the Web Services Interoperability (WS-I) in 2002. The organization is chartered to promote Web Services interoperability across platforms, applications, and programming languages. Microsoft has managed to create an expectation that Microsoft products will conform to Web Services standard. IT Vendors, especially Microsoft’s partners, are comfortable to use Microsoft products without fear of being incompatible with other vendors’ systems in providing Web Services, and this retains the installed base of Microsoft platforms that are already built huge.

Second, Microsoft augments modules of .NET Passport, MSN and Windows Messenger, and Microsoft Alerts as part of the Web Services platform. .NET Passport is a single sign-in service for the customer to manage her personal identification on the Internet, and MSN or Windows Messenger is the software for people with .NET Passports to communicate via text, voice, and video. Microsoft Alerts are brief, time-sensitive, action-oriented messages that content or service providers can send to users on the users’ requests. The more popular are the .NET passport, Messenger, the more Web sites are willing to connect customers with Microsoft Alerts and willing to adopt Microsoft solutions. This is a direct network effect cycle resulted from the company’s design of platform.

Third, Microsoft provides tools and building blocks of Web Services, saving the cost of assembling and providing application services. The broad installed base of Microsoft platforms increases the potential value of providing application services. Theoretically, this will lead to more variety of Web Services available in the market. However, because Microsoft often competes with complementors in complementary innovations, the variety of Microsoft enabled application services tends not to increase much. Microsoft’s business model on this cycle is primarily selling Web Services technologies to small and medium-sized business customers and directly profit from that. These business customers may transform their conventional businesses into niche Web Services and provide them online.

The positive-feedback cycles are illustrated below, and the dot line represents a weak or broken causal relation:
FIGURE 2 Positive-feedback cycles of Microsoft .NET Strategy

CONCLUSION

The analysis of Microsoft .NET strategy shows that the company is developing its Web Services platform depending on conditions of direct and indirect network effects. Modules of direct network effect are gradually developed and integrated into the Web Services platform. In order to bring about direct network effect, Microsoft is building MSN into a transaction platform made of Web Services platforms to directly link to end-users, and major Microsoft products may become fronts for end-users to transact services with MSN or the company.

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

For the platform provider, modularization and demodularization decisions should be made according to triggering conditions of network effect. These conditions are 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 changing system modularity to adjust these conditions to start positive-feedback cycles for the platform provider to create value and appropriate rent.

Complementary modules with demand-side economies of scale should be integrated into the platform module. Because such modules may boost the market demand and entail high switching cost to end-users, once the collective switching cost gets high, the complementary
module provider will gain power over the platform provider and may become a competitor if the platform is not difficult to produce. The platform provider should demodularize the system by integrating modules with such potential. While pursuing indirect network effect, the platform provider should confine complementors to contribute to the adoption of the system product only through the increase of component variety and decrease of price. If large component variety is not necessary for system product adoption, the platform provider should resort to the approach of direct network effect if possible. With the author’s framework, product designers and strategy makers can assess the network context to formulate their platform strategy in a new perspective.

REFERENCES


